inubit Suite 6

inubit Workbench/Process Engine

Modul-Guide: Data Converter, Format Adapter, Utilities, Workflow and Web Service Controls

Copyright

inubit AG

Schöneberger Ufer 89-91
10785 Berlin
Deutschland

Phone: +49.30.72 61 12-0
Fax: +49.30.72 61 12-100
E-Mail: contact@inubit.com
URL: www.inubit.com

© inubit AG 2011

Rechtliche Bestimmungen

Die in diesen Unterlagen enthaltenen Angaben und Daten, einschließlich URLs und anderer Verweise auf Internetbasis, können ohne vorherige Ankündigung geändert werden. Die Produktdokumentation wurde sorgfältig erstellt. Die darin enthaltenen Angaben können jedoch nicht als Zusicherung von Eigenschaften der inubit Suite 6 gelten. Die Haftung der inubit AG umfasst nur die in den Verkaufs- und Lieferbedingungen festgelegten Bestimmungen.

Die Benutzer sind verantwortlich für das Einhalten aller anwendbaren Urheberrechtsgesetze. Unabhängig von der Anwendbarkeit der entsprechenden Urheberrechtsgesetze darf ohne ausdrückliche schriftliche Erlaubnis der inubit AG kein Teil dieses Dokuments für irgendwelche Zwecke vervielfältigt oder übertragen werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln, elektronisch oder mechanisch, dies geschieht.

Die inubit AG kann Inhaber von Patenten, Marken, Urheberrechten oder anderen Rechten an geistigem Eigentum sein, die den Inhalt dieses Dokuments betreffen. Die Bereitstellung dieses Dokuments erteilt keinerlei Lizenzrechte an diesen Patenten, Marken, Urheberrechten oder anderem geistigen Eigentum, ausgenommen, dies wurde explizit durch einen schriftlich festgelegten Lizenzvertrag mit der inubit AG vereinbart.

Die von der inubit AG angebotene Software kann Softwarekomponenten anderer Hersteller enthalten. inubit ist eine eingetragene Marke der inubit AG. Alle sonstigen in diesem Dokument aufgeführten Produkt- und Firmennamen sind möglicherweise eingetragene Marken der jeweiligen Inhaber.

Inhaltsverzeichnis

Hinweise zum Modul-Guide 11
Umfang der Dokumentation 11
Tipps, Hinweise und Links in der Dokumentation 13
Weitere Informationen und Support 13

1 Complex Lookup Table (Data Converter) 15
1.1 Funktionsprinzip 15
1.2 Verarbeitungsvorschriften anlegen 17
1.3 Beispiel: Elemente erstellen 20
1.4 Beispiel: Feste Werte als zusätzliches Suchkriterium 22
1.5 Beispiel: Variablenwert als zusätzliches Suchkriterium 24
1.6 Beispiel: XPath-Ausdruck als zusätzliches Suchkriterium 27
1.7 Dialogbeschreibungen 28
1.7.1 Register „Datenquellen“ 28
1.7.1.1 Dialog „Datenquelle hinzufügen“ (1) 29
1.7.1.2 Dialog „Datenquelle hinzufügen“ (2) 30
1.7.2 Register „Verarbeitungsvorschriften“ 31
1.7.2.1 Dialog „Eigenschaften der Verarbeitungsvorschrift“ 33
1.7.2.2 Bereich „Aktionsliste“ 34

2 FO Converter (Data Converter) 37
2.1 PDF/A-1b erstellen und Fonts ändern 37
2.2 UNIX/Linux: SVG-Grafiken erzeugen 40
2.3 Dialog „FO Converter Eigenschaften“ 40

3 Lookup Table (Data Converter) 43
3.1 Funktionsprinzip 43
3.2 Datenquelle anlegen 45
3.2.1 XML-Datenquelle definieren 46
3.2.2 Daten in eine XML-Datenquelle importieren 46
3.2.3 Daten manuell zu einer XML-Datenquelle hinzufügen 47
3.2.4 Daten einer XML-Datenquelle bearbeiten und löschen 48
3.2.5 DB-Datenquelle definieren 48
3.3 Datenbanktabelle referenzieren, Tabellenname und Spaltennamen setzen 49
3.4 Tabellen und Spalten mit dem DB-Manager definieren 50

3.5 Ersetzungsvorschriften definieren 51
3.6 Dialogbeschreibungen 52
3.6.1 Register „Datenquellen“ 52
3.6.1.1 Dialog „Datenquelle hinzufügen“ (1) 53
3.6.1.2 Dialog „Datenquelle hinzufügen“ (2) 53
3.6.1.3 Dialog „Tabelleneigenschaften“ 54
3.6.2 Register „Verarbeitungsvorschriften“ 55
3.6.2.1 Dialog „Eigenschaften der Verarbeitungsvorschrift“ 56

4 Task Generator (Data Converter) 59
4.1 Funktionsprinzip des Task Generators 60
4.2 Formulare im Formular Designer erstellen 63
4.2.1 Formular-Layout festlegen und Elemente auf Panels positionieren 63
4.2.2 Formularvorschau anzeigen 65
4.2.3 Formulare mit CSS gestalten 65
4.2.3.1 CSS zentral definieren 66
4.2.3.2 CSS lokal definieren 66
4.2.3.3 Alternativen 67
4.2.4 Combo-Boxen erstellen 68
4.2.5 Liste mit Mehrfachauswahl dynamisch erstellen 70
4.2.6 Maßeinheit für Größenangaben ändern 71
4.2.7 Formularelemente abhängig von Rechten machen (nur bei Web-Applikationen) 71
4.2.8 Formularteile aus XML Schemas/WSDL-Dateien mit dem Client Generator erzeugen 72
4.2.9 JavaScript in Formularen verwenden 74
4.2.10 Portlets und Formularabschnitte mit Ajax aktualisieren 76
4.2.10.1 Portlet mit Ajax aktualisieren 77
4.2.10.2 Formularabschnitt mit Ajax aktualisieren 78
4.3 Benutzereingaben in Formularen validieren 80
4.3.1 Clientseitige Validierung von Benutzereingaben über Pattern 81
4.3.2 Server-seitige Validierung von Benutzereingaben per AJAX 82
4.4 Kontextsensitive Onlinehilfe erstellen 84
4.5 Formular-Texte zum Übersetzen exportieren und importieren 87
4.6 Widgets verwenden und erstellen 90
4.6.1 Grid-Widget verwenden und konfigurieren 91
4.6.1.1 Beispiel: Dynamische Tabelle mit client-seitiger Sortierung, ohne Paging 93
4.6.1.2 Beispiel: Dynamische Tabelle mit Paging und server-seitiger Sortierung 94
4.6.2 Widget erstellen und in die inubit Suite 6 integrieren 97

4.7 Mitarbeiter mit Tasks in Workflows einbeziehen 100
4.7.1 Tasks in Technical Workflows integrieren 103
4.7.2 Tasklisten anzeigen 105
4.7.2.1 Taskliste in der inubit Workbench anzeigen 106
4.7.2.2 Tasklist als Servlet im Browser anzeigen 106
4.8 Benutzeroberfläche des Formular Designers 107
4.8.1 Formular Mapping 108
4.8.2 Ausgangsmapping 110
4.8.3 Formular-Elemente in der Werkzeugleiste 111
4.8.4 Eigenschaften von Formular-Elementen 114
4.9 Benutzeroberfläche des Client Generators 120
4.10 Dialogbeschreibungen 124
4.10.1 Dialog „Task Generator Eigenschaften“ 124
4.10.2 Dialog „Berechtigungen“ 127
4.10.3 Dialog „Applikationsprofil“ 128
4.10.4 Dialog „XSLT Mapper Einstellungen“ 129
4.10.5 Dialog „Panel Layout“ 130

5 XSLT Converter (Data Converter) 133
5.1 Benutzeroberfläche: XSLT Mapper 134
5.1.1 Funktionen in der Werkzeugleiste 136
5.1.2 Weitere Funktionen 139
5.1.3 XSLT-Kommandoassistent 140
5.2 XSLT-Stylesheets testen 142
5.3 XSLT-Stylesheets debuggen 142
5.4 Ergebnis validieren 143
5.5 Extensions und Java-Packages verwenden 143
5.5.1 Extensions verwenden 144
5.5.2 Java-Packages der inubit Suite 6 einbinden 145
5.6 Beliebige XSLT-Prozessoren einbinden 145
5.7 Variablen verwenden 146
5.8 Dialog „XSLT Converter Eigenschaften“ 148

6 CSV Adapter (Format Adapter) 149
6.1 Beispiel: Konvertierung CSV nach XML 150
6.2 Register „Adapter Editor“ 151

7 EDI Adapter (Format Adapter) 155
7.1 Funktionsprinzip der EDI>XML-Konvertierung 156
7.2 EDI-XML Adapter erstellen und Regel konfigurieren 158
7.3 XML-Eingangsnachricht für XML-EDI Adapter erstellen 160
7.4 Neue Regel erstellen 161
7.5 Regeln zur Wiederverwendung auf die inubit Process Engine publizieren 162
7.6 SAP IDoc XML-Regel erstellen 163
7.7 Regeleditoren verwenden 164
7.8 Dialogbeschreibungen 166
7.8.1 Dialog „Qualifierdefinition“ 167
7.8.2 Dialog „EDI/XML-Adaptereigenschaften“ 168
7.8.3 Dialog „Syntax und Validierung“ 170
7.8.4 Dialog „Trennzeichen“ 172
7.8.5 Dialog „XML/EDI-Adaptereigenschaften“ 173
7.8.6 Assistent für neue Regel: Erstellungsmethode auswählen 174
7.8.7 Assistent für neue Regel: Neue Regel 174

8 Flat Adapter (Format Adapter) 177
8.1 Funktionsprinzip 177
8.2 Regel-Editor (Oberfläche) 180
8.3 Neue Regel erstellen 182
8.4 Dialogbeschreibungen 183
8.4.1 Dialog „Flatfile-XML Adaptereigenschaften“ 183
8.4.2 Dialog „XML-Flatfile Adapter Eigenschaften“ 185

9 MIME Adapter (Format Adapter) 189
9.1 Beispiel: Konvertierung MIME > XML-MIME 190
9.2 Dialogbeschreibungen 191
9.2.1 Dialog „MIME-XML Adaptereigenschaften“ 192
9.2.2 Dialog „XML-MIME Adaptereigenschaften“ 192

10 XLS Adapter (Format Adapter) 193
10.1 Eingangsnachrichten für XML-XLS Adapter erstellen 193
10.2 XSL > XML: Zeitzone zum Konvertieren von Datumswerten umstellen 194
10.3 Dialogbeschreibungen 194
10.3.1 Dialog „XLS-XML Adapter Eigenschaften“ 195
10.3.2 Dialog „XML-XLS Adapter Eigenschaften“ 196

11 XLSX Adapter (Format Adapter) 199
11.1 Eingangsnachrichten für die Konvertierung von XML > XLSX erstellen 199
11.2 Dialog „XLSX Adapter Eigenschaften“ 200

12 Barcode Generator (Utility) 201
12.1 Beispiel-Workflow 201
12.2 Dialog „Barcode Generator Eigenschaften“ 203

13 Compressor (Utility) 205
13.1 Zip-Archiv mit mehreren Dateien erstellen 205
13.2 Dialog „Compressor/Decompressor Eigenschaften“ 206

14 Cryptographer (Utility) 209
14.1 Dialogbeschreibungen 209
14.1.1 Dialog „Cryptographer Plug-in“ 209
14.1.2 Dialog „Websale Cryptographic 3DES“ 210
14.1.3 Dialog „CMS (Cryptographic Message Syntax)“ 210

15 Data Stream Modifier (Utility) 213
15.1 Dialog „Data Stream Modifier Eigenschaften“ 213

16 Encoder (Utility) 215
16.1 Dialog „Encoder Eigenschaften“ 215

17 inubit IS Configuration (Utility) 217
17.1 Funktionsprinzip 218
17.2 Eingangsnachricht zum Aufrufen der Operationen erstellen 219
17.3 Verschlüsselte Login-Daten verwenden 220
17.4 Operationen remote/als anderer Benutzer ausführen 221
17.5 Logeinträge aus Monitoring-Komponenten lesen 221
17.6 Logeinträge filtern 224
17.7 Logeinträge sortieren 225
17.8 In Logeinträgen nach Zeichenketten suchen 226
17.9 Größe von Eingangsnachrichten ermitteln 226

18 Line Reader (Utility) 229
18.1 Dialog „Line Reader Eigenschaften“ 229

19 PDF Toolbox (Utility) 231
19.1 Eingangsnachricht für PDF Toolbox erzeugen 231
19.2 Beispiel-Workflow 232

20 Process Data Logger (Utility) 235
20.1 Process Data Logger konfigurieren 235
20.2 Dialog „Process Data Logger Eigenschaften“ 236

21 Report Data Collector (Utility) 239
21.1 Einsatz unter UNIX/Linux 239
21.2 Prozessmodell-Report erstellen 240
21.3 Prozessmodell-Report anpassen 240
21.4 Dialogbeschreibungen 242
21.4.1 Dialog „Diagrammauswahl“ 242
21.4.2 Dialog „Konfiguration der Auswahl“ 242

22 S/MIME (Utility) 245
22.1 Schlüssel und Zertifikate erstellen 245
22.2 Dialogbeschreibungen 246
22.2.1 Dialog „S/MIME-Konfiguration“ 246

23 XML Diff (Utility) 247
23.1 Ausgangsnachrichten 247
23.2 Dialog „XML Diff Moduleinstellungen“ 248

24 XML Enveloper (Utility) 249
24.1 Dialog „XML Enveloper Eigenschaften“ 249

25 XML Formatter (Utility) 251
25.1 Beispiel für Ein- und Ausgangsnachrichten 251
25.2 Dialog „XML Formatter Eigenschaften“ 252

26 XML Validator (Utility) 253
26.1 Dialog „XML Validator Eigenschaften“ 253

27 Web Service Controls 255
27.1 Invoke 255
27.1.1 Dialog beschreibung 255

27.2 OnMessage 257
27.3 Receive 257
27.3.1 Dialog „Web Service Einstellungen“ 258
27.4 Reply 259
27.4.1 Dialogbeschreibung 259
27.4.1.1 Dialog „Reply-Moduleigenschaften“ in Technical Workflows 259
27.4.1.2 Dialog „Reply-Moduleigenschaften“ in BPEL-Diagrammen 260

28 Workflow Controls 261
28.1 Assign 261
28.2 Compensate 262
28.3 Compensate Scope 262
28.4 Demultiplexer 263
28.4.1 Bedingungen für die Nachrichtenweiterleitung definieren 263
28.4.2 Dialog „Demultiplexer Konfiguration“ 265
28.5 Empty 267
28.6 Exit 267
28.7 Joiner 268
28.8 Multiplexer 268
28.8.1 Dialog „Multiplexer Eigenschaften“ 269
28.9 Rethrow 272
28.10 Splitter 272
28.10.1 Dialog „Splitter“ 272
28.11 Throw 273
28.11.1 Dialog „Throw-Einstellungen“ 273
28.12 Validate 274
28.12.1 Dialog „Validate-Moduleinstellungen“ 274
28.13 Wait 274
28.14 Workflow Connector 275
28.14.1 Dialogbeschreibungen „Workflow Connector“ 276
28.14.1.1 Dialog „Workflow-Auswahl und Eigenschaften“ 276
28.14.1.2 Dialog „Modul-Auswahl“ 278
28.15 Workflow Starter 278
28.15.1 Dialog „Workflow Starter“ 279

Zielgruppe

Der Modul-Guide enthält detaillierte Informationen für Systemadministratoren, Systemintegratoren und Entwickler, welche die inubit Suite 6 (oder Teile davon) verwalten und konfigurieren.

In diesem Guide werden folgende Module erläutert:

- **Data Converter**
  Data Converter konvertieren Daten, die in einem XML-Format vorliegen, in ein anderes XML-Format.
- **Format Adapter**
  Format Adapter überführen Daten, die nicht in einer XML-Struktur vorliegen (z. B. CSV- oder EDI-Datenformate), in XML-Formate.
- **Utilities**
  Utilities sind nützliche Zusatzmodule, um die Abarbeitung eines Technical Workflows zu optimieren. Mit den Utilities können Sie z. B. Daten komprimieren/dekomprimieren, kodieren/dekodieren oder Prozessdaten protokollieren.
- **Workflow Controls**
  Workflow Controls sind Kontrollelemente, welche die Abarbeitungsreihenfolge von Modulen in Technical Workflows beeinflussen (Iterationen, Sequenzen, bedingte Auswahl etc.)
- **Web Service Controls**
  Web Service Controls sind in BPEL-Prozessen die zentralen Elemente zum Aufrufen und Anbieten von Web Services. In Technical Workflows können Sie Web Service Controls als Alternative zu Web Services Connectoren einsetzen.

Umfang der Dokumentation

Die inubit Suite 6 bietet eine umfassende Dokumentation, die Ihnen als gedrucktes Handbuch, als PDF-Datei und als Onlinehilfe in inubit Workbench zur Verfügung steht.

Die Dokumentation besteht aus folgenden Teilen:

- **inubit Suite 6**
  - **Quick Start**
    Beschreibt die Hard- und Softwarevoraussetzungen, die Installation und die ersten Schritte.
  - **Migrationsanleitung**
  - **Tutorials**
    Für den Ein- und Durchstieg. Die Tutorials erläutern die Verwendung der wichtigsten Komponenten der inubit Suite 6 anhand von fachlichen Szenarien.
- **inubit Workbench, inubit Process Engine** und **inubit Enterprise Portal**:

- **Benutzer-Guide**
  Beschreibt das Arbeiten mit inubit Workbench, das Erstellen der verschiedenen Diagrammtypen und Module, das Arbeiten mit Metadaten und Workflow Variablen, Simulationen, Tests, das fachlich orientierte Monitoring und Reporting.
- **Administrator- und Entwickler-Guide**
  Enthält administrative Themen wie Konfiguration der inubit Process Engine, Backup und Restore, Benutzerverwaltung, Security-Aspekte, Monitoring und Clustering, Entwicklung von Plug-ins und Thin Clients.
- **Modul-Guide**
  Data Converter, Format Adapter, Utilities, Workflow und Web Service Controls verwenden und konfigurieren.
- **Systemkonnektor-Guide**
  Alles über Einsatz und Konfiguration von Systemkonnektoren.

- **inubit Solution Center 3.0**
  - **Benutzer-Guide**
    Informationen, wie Sie Fachmodelle anlegen, mit Business Solutions arbeiten, Ansichten erstellen, Prozesse und Standardmodelle einbinden sowie eine Dokumentation der REST-Schnittstelle.
  - **Administrator-Guide**
    Erläutert, wie Sie Daten sichern und wiederherstellen, das inubit Solution Center Server als Dienst installieren, Ports anpassen, das Portal, die Datenbank und HTTPS konfigurieren, Benutzer verwalten und Diagramme importieren.
- **inubit WebModeler 2.3**
  - **Administrator- und Benutzer-Guide**
    Alles über das Anlegen und Bearbeiten von Modellen, Anpassen der Ports, Konfigurieren der Datenbank und von HTTPS.

Die aktuelle Dokumentation steht im inubit User-Portal im Register „Software“ unter der Adresse https://www.inubit-user.com zum Download bereit.

Weitere Informationen

Die folgenden Informationen liegen als Booklet der DVD bei bzw. sind als Dateien im Installationspaket enthalten:

- **readme.txt**
  Hinweise zur Installation und Migration der inubit Suite 6.

Lesen Sie diese Datei grundsätzlich vor der Installation oder Aktualisierung der inubit Suite 6!

- **Quick Start**

Systemvoraussetzungen und Installationsanleitung als Booklet der inubit Suite 6-DVD.

- **API-Dokumentation des Plug-in Software Development Kits**
  Im Verzeichnis <`iS`-installdir>/documentation/apidoc/index.html.
- **JavaScript-Framework**
  Im Verzeichnis `<iS-installdir>/documentation/jsdoc/index.html`.

Tipps, Hinweise und Links in der Dokumentation

Tipps bieten nützliche Informationen für das Arbeiten mit der inubit Suite 6.

Hinweise sollten Sie unbedingt lesen und beachten. Das Nichtbeachten kann den Verlust von Daten oder schwerwiegende Systemprobleme verursachen.

→ Verweise auf eine andere Textstelle in der Dokumentation der inubit Suite 6 sind mit einem Pfeil gekennzeichnet.

Links auf Webseiten erkennen Sie an dem nebenstehenden Symbol.

Weitere Informationen und Support

Pressemitteilungen und Whitepapers stehen auf unserer Website www.inubit.com für Sie zum Download bereit.

Für weitere Informationen über die inubit Suite 6 oder bei Fragen zum Einsatz der inubit Suite 6 nutzen Sie folgende Kontaktmöglichkeiten:

- **inubit-Support**: www.inubit.com/support
- **E-Mail**: support@inubit.com
- **Telefon**: +49.30.72 61 12-112

Viel Erfolg beim Arbeiten mit der inubit Suite 6 wünscht Ihnen das inubit Team!

1 Complex Lookup Table (Data Converter)

Dieser Abschnitt erläutert die folgenden Themen:

- *Funktionsprinzip, S. 15*
- *Verarbeitungsvorschriften anlegen, S. 17*
- *Beispiel: Elemente erstellen, S. 20*
- *Beispiel: Feste Werte als zusätzliches Suchkriterium, S. 22*
- *Beispiel: Variablenwert als zusätzliches Suchkriterium, S. 24*
- *Beispiel: XPath-Ausdruck als zusätzliches Suchkriterium, S. 27*
- *Dialogbeschreibungen, S. 28*

Verwendung

Mit einer Complex Lookup Table können Sie Werte in XML-Nachrichten auf Basis komplexer Verarbeitungsvorschriften ersetzen lassen und Elemente zu den XML-Nachrichten hinzufügen oder Elemente entfernen.

1.1 Funktionsprinzip

Verarbeitungsvorschriften und Aktionen

Jedes Modul kann mehrere Verarbeitungsvorschriften enthalten. Eine Verarbeitungsvorschrift besteht aus mehreren Aktionen, die auf den Eingangsnachrichten durchgeführt werden, wie z. B. dem Suchen nach einem bestimmten Wert und dem anschließenden Ersetzen dieses Wertes.

Die zu ersetzenden Werte und die Ersatzwerte werden in Tabellen vorgehalten und aufeinander abgebildet. Als Tabellen können Sie statische XML-Tabellen und dynamische Datenbanktabellen verwenden:

- Statische Tabellen werden direkt in der Complex Lookup Table angelegt und mit den Wertepaaren gefüllt.
- Dynamische Tabellen können aus einer Datenbank geladen werden.

Stapelverarbeitung

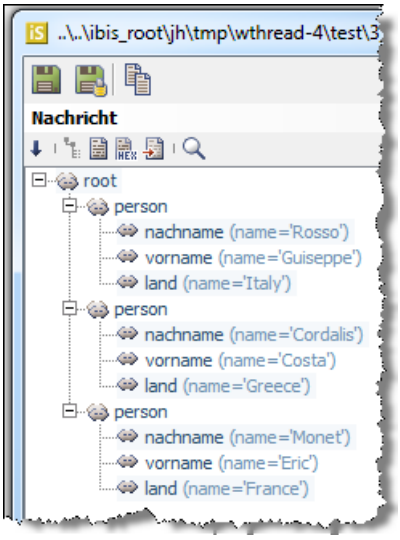
Die Verarbeitungsvorschriften werden in der Reihenfolge ihres Vorkommens abgearbeitet. Jede Verarbeitungsvorschrift wird genau einmal angewendet.

Die Complex Lookup Table ruft die erste Verarbeitungsvorschrift auf und verarbeitet damit die Eingangsnachricht. Danach wird die nächste Verarbeitungsvorschrift auf die bereits veränderte Nachricht angewendet. Die Eingangsnachricht verändert sich schrittweise mit jeder Verarbeitungsvorschrift.

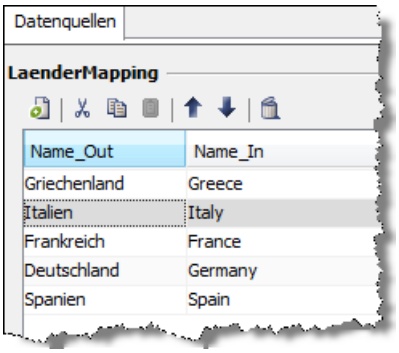
Beispiel: Werte ersetzen

In diesem Beispiel werden englische Ländernamen durch deutsche ersetzt.

- Die Eingangsnachricht hat folgende Struktur:

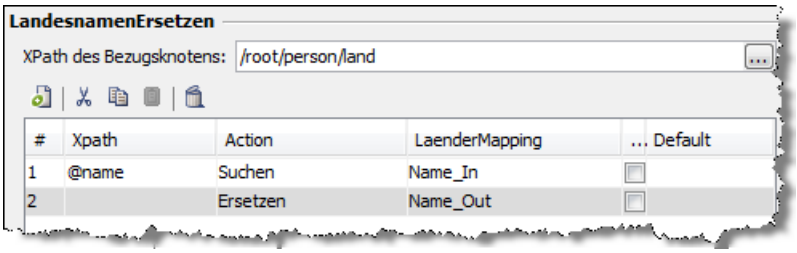


- Die **Verarbeitungsvorschrift** lautet:
  a. Suche nach dem Wert des ersten `land`-Elements, das Kindelement von `/root/person` ist und dessen Wert in der Tabelle `LaenderMapping` in der Spalte `Name_In` steht.
  Hier: `Italy`.
  b. Ersetze den vorhandenen Wert mit dem Wert, der in derselben Tabelle zu finden ist - und zwar in der Spalte `Name_Out` in derselben Zeile wie `Italy`. Hier ist das `Italien`:

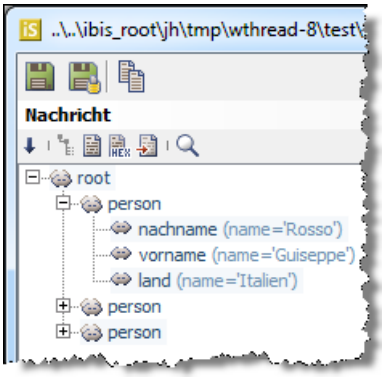


  c. Verfahre genauso mit allen weiteren Werten der `land`-Elemente.

  In der Complex Lookup Table wird diese Vorschrift folgendermaßen formuliert:



- Die **Ausgangsnachricht** sieht so aus:



→ Siehe *Verarbeitungsvorschriften anlegen (Workbench/Process Engine: Modul-Guide, Kap. 1.2, S. 17).*

1.2 Verarbeitungsvorschriften anlegen

Voraussetzungen

- Sie benötigen eine XML-basierte Beispielnachricht mit derselben Struktur wie die Eingangsnachrichten, die im Produktivbetrieb die Complex Lookup Table durchlaufen werden.
- Wenn Sie eine Datenbank als Datenquelle nutzen möchten:
  - Der zur Datenbank passende Datenbanktreiber muss installiert sein.
    → Siehe *Treiber installieren (Process Engine: Administrator- und Entwickler-Guide, Kap. 3.4, S. 51).*
  - Sie benötigen die Zugangsdaten (Benutzername und Passwort) zu der Datenbank.

So gehen Sie vor

1. **Datenquelle erstellen**
   a. Zeigen Sie das Register „Datenquellen“ an.

b. Öffnen Sie im linken Bereich das Kontextmenü und wählen Sie „Hinzufügen“. Der *Dialog „Datenquelle hinzufügen“ (1) (Workbench/Process Engine: Modul-Guide, Kap. 1.7.1.1, S. 29)* öffnet sich.

c. Aktivieren Sie die Option „XML-Datenquelle“ oder „DB-Datenquelle“.

d. Geben Sie einen eindeutigen Namen und optional eine Beschreibung für die neue Datenquelle an.

e. **Nur bei DB-Datenquellen**

Klicken Sie auf „Weiter“. Der *Dialog „Datenquelle hinzufügen“ (2) (Workbench/Process Engine: Modul-Guide, Kap. 1.7.1.2, S. 30)* öffnet sich. Konfigurieren Sie die Verbindung zur Datenbank.

f. Klicken Sie auf „Fertig stellen“.

Ihre Datenquelle wird im linken Bereich angezeigt. XML-Datenquellen sind durch das Präfix „XML:“, DB-Datenquellen durch das Präfix „DB:“ gekennzeichnet.

2. **Daten in der Datenansicht eingeben/anzeigen**

- **XML-Datenquellen**

Sie können Daten importieren oder manuell eingeben:

  - Zum Importieren öffnen Sie im rechten Bereich das Kontextmenü und wählen „Tabellendaten importieren“.
  - Um Daten manuell einzugeben, öffnen Sie im rechten Bereich das Kontextmenü und wählen „Hinzufügen“.

- **DB-Datenquelle**

a. Öffnen Sie im rechten Bereich das Kontextmenü und wählen Sie „Tabelleneigenschaften“. Ein Dialog öffnet sich.

b. Wählen Sie die Datenbanktabelle aus und aktivieren Sie die Checkbox der Spalten, die Sie mit der Verarbeitungsvorschrift bearbeiten möchten.

c. Klicken Sie auf „OK“, um die Auswahl zu speichern. In der Datenansicht erscheint eine leere Tabelle mit den Spaltennamen als Überschrift.

d. Um die Daten aus der Datenbank in die Spalten zu laden, wählen Sie aus dem Kontextmenü „Hinzufügen“.

Die ersten 25 Datensätze werden aus der Datenbank gelesen und angezeigt.

3. **Verarbeitungsvorschrift anlegen**

a. Zeigen Sie das Register „Verarbeitungsvorschriften“ an.

b. Öffnen Sie im linken Bereich das Kontextmenü und wählen Sie „Hinzufügen“. Der *Dialog „Eigenschaften der Verarbeitungsvorschrift“ (Workbench/Process Engine: Modul-Guide, Kap. 1.7.2.1, S. 33)* öffnet sich.

c. Benennen Sie die Verarbeitungsvorschrift. Der Name wird in der Liste der Verarbeitungsvorschriften und als Überschrift der Aktionsliste angezeigt. Setzen Sie die übrigen Parameter und wählen Sie die zu benutzenden Datenquellen aus.

d. Klicken Sie auf „OK“, um die Verarbeitungsvorschrift zu speichern.

4. **Beispielnachricht laden**

a. Klicken Sie im Bereich „Quelldokument“ auf den -Button und wählen Sie „Öffnen“. Ein Dateiexplorer öffnet sich.

b. Navigieren Sie zur Beispielnachricht und laden Sie diese. Die geladene Nachricht wird in demselben Bereich angezeigt.

5. **XPath des Bezugsknotens setzen**

In jeder Verarbeitungsvorschrift müssen Sie einen gemeinsamen Bezugsknoten für alle Aktionen mit XPath festlegen. Sie können den XPath manuell eingeben oder das Element aus der Beispielnachricht per Drag’n’Drop auf das Eingabefeld „XPath des Bezugsknotens“ ziehen.

6. **Aktionen definieren**

Die erste Aktion muss immer eine Suche nach einem Element sein. Wenn die Suche erfolgreich war, dann können Sie dessen Werte ersetzen oder neue Elemente erstellen lassen.

- **Suchen**

  Für die Suche nach einem Element können Sie mehrere Bedingungen festlegen. Alle Bedingungen sind implizit mit logischem „UND“ verbunden.

  Wenn eine Suchbedingung nicht zutrifft, dann wird die nächste Bedingung geprüft (wenn vorhanden). Wenn keine weitere Bedingung vorhanden ist, dann werden folgende Aktionen durchgeführt:

  - Die nächste Verarbeitungsvorschrift wird ausgeführt (wenn vorhanden).
  - Wenn nicht, dann wird für die Ersetzung der Standardwert verwendet (wenn vorhanden).
  - Wenn nicht, dann wird die Verarbeitung des Moduls abgebrochen und eine Fehlermeldung angezeigt.

  Die erste Suche muss immer auf einer Tabelle stattfinden. Die zweite Suche kann sich auf Werte in den Eingangsnachrichten beziehen oder auf zusätzliche feste Werte, wenn die Spaltenangabe nicht ausreicht, um eine Zeile eindeutig zu identifizieren, oder wenn ausschließlich nach Werten aus einer bestimmten Zeile gesucht werden soll.

  Darüber hinaus können Sie auch nach Workflow-Variablen suchen. Dazu geben Sie Namen der gewünschten Variablen mit einem vorangestellten $-Zeichen in der Spalte „Fest“ ein.

  → Siehe

- *Beispiel: Feste Werte als zusätzliches Suchkriterium (Workbench/Process Engine: Modul-Guide, Kap. 1.4, S. 22)*
- *Beispiel: Variablenwert als zusätzliches Suchkriterium (Workbench/Process Engine: Modul-Guide, Kap. 1.5, S. 24)*
- *Beispiel: XPath-Ausdruck als zusätzliches Suchkriterium (Workbench/Process Engine: Modul-Guide, Kap. 1.6, S. 27)*

Wenn in einer Datenquelle gesucht werden soll, dann darf der XPath-Ausdruck nicht mit eckigen Klammern enden, wie z. B. `book[@id='20']`. Nutzen Sie stattdessen die XSLT-Funktion `text()`, z. B. `book[@id='20']/text()`.

- **Ersetzen**
  Das Element, dessen Wert ersetzt werden soll, wird über einen XPath adressiert. Das Element muss unterhalb des Bezugsknotens liegen.
  → Siehe *XPath des Bezugsknotens setzen (Workbench/Process Engine: Modul-Guide, Kap. 1, S. 19)*.
- **Erstellen**
  Es wird ein XML-Element erstellt und diesem ein Wert zugewiesen. Die Position und der Name des neuen Elements werden über einen XPath angegeben, z. B. erzeugt `/root/person/Ort` ein Element `Ort` unterhalb des XML-Knotens `person`. Das Element muss unterhalb des Bezugsknotens liegen.
  → Siehe *XPath des Bezugsknotens setzen (Workbench/Process Engine: Modul-Guide, Kap. 1, S. 19)*.

Wenn Sie eine Suche formuliert haben und danach eine Erstellung und/oder Ersetzung konfiguriert haben, können Sie keine weitere Aktion in dieser Verarbeitungsvorschrift definieren. Für weitere Aktionen müssen Sie eine neue Verarbeitungsvorschrift erstellen.

7. **Verarbeitungsvorschrift testen**

   Klicken Sie in der Werkzeugleiste auf den -Button.

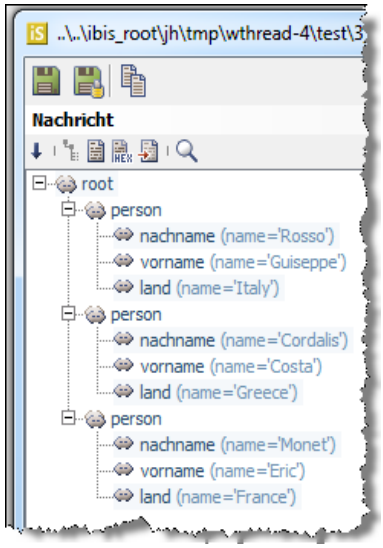
Das Testergebnis wird im Bereich „Ergebnis“ angezeigt. Falls Fehler auftreten, werden diese als Meldung ausgegeben.

1.3 Beispiel: Elemente erstellen

Das Beispiel erläutert das Erstellen von Elementen.

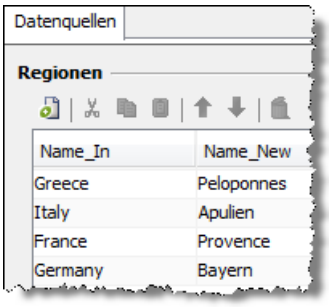
Eingangsnachricht

Die Eingangsnachricht hat folgende Struktur:



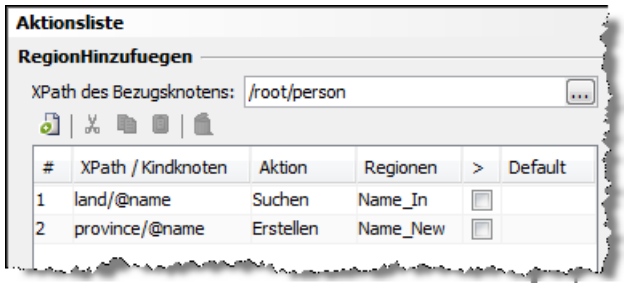
Verarbeitungsvorschrift

1. Suche nach dem ersten `land`-Element, das Kindelement von `/root/person` ist und dessen Wert in der Datenquelle `Regionen` in der Spalte `Name_In` steht.
   Hier: `Italy`.
2. Erstelle das Element `province` mit dem Wert, der in derselben Datenquelle zu finden ist - und zwar in der Spalte `Name_New` in derselben Zeile wie `Italy`. Hier ist das `Apulien`:



Name_In	Name_New
Greece	Peloponnes
Italy	Apulien
France	Provence
Germany	Bayern

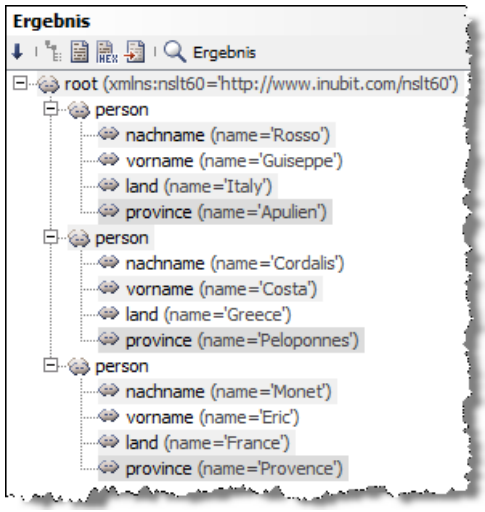
In der Aktionsliste sieht diese Vorschrift so aus:



#	XPath / Kindknoten	Aktion	Regionen	>	Default
1	land/@name	Suchen	Name_In		
2	province/@name	Erstellen	Name_New		

Ausgangsnachricht

Das Ergebnis ist folgendes:



Um sicherzustellen, dass auch dann neue Elemente erstellt werden, wenn in einer Datenquelle Werte fehlen, geben Sie einen Default-Wert an.

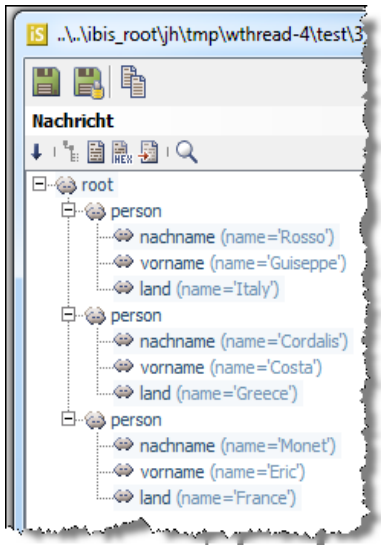
1.4 Beispiel: Feste Werte als zusätzliches Suchkriterium

Das Beispiel erläutert, wie Sie feste Werte als zusätzliche Suchkriterien verwenden. Dies ist nötig, wenn die Spaltenangabe nicht ausreicht, um eine Zeile eindeutig zu identifizieren, oder wenn ausschließlich nach Werten aus einer bestimmten Zeile gesucht werden soll.

Im Beispiel soll der aktuelle Wert des Elements `land` durch den Wert ersetzt werden, der in der Spalte `Name_New` steht. Allerdings ist die Zuordnung der Ausgabewerte zu den Eingabewerten nicht eindeutig, da einige Werte in der Datenquelle mehrfach vorkommen und unterschiedlichen Werten zugeordnet sind. Als zusätzliches Kriterium gilt daher: Es soll genau der Wert ausgegeben werden, bei dem in der Spalte `Season` ein ganz bestimmter Wert steht.

Eingangsnachricht

Die Eingangsnachricht hat folgende Struktur:

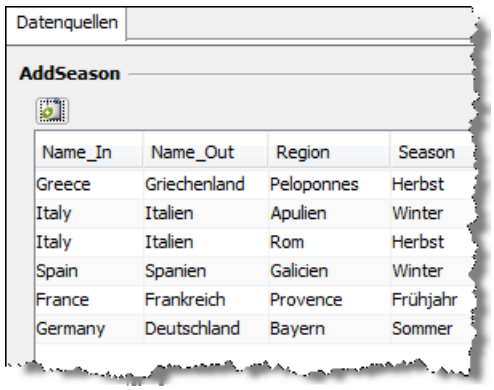


Verarbeitungsvorschrift

Im Bereich „Lookup Konfiguration“ wird die Option „Feste Werte als zusätzliches Suchkriterium benutzen“ markiert. Die Spalte „Fest“ wird angezeigt.

Die folgenden Aktionen werden erstellt:

1. Suche in der Datenquelle `AddSeason` in der Spalte `Name_In` nach dem Wert des ersten Elements `land` aus der Eingangsnachricht.

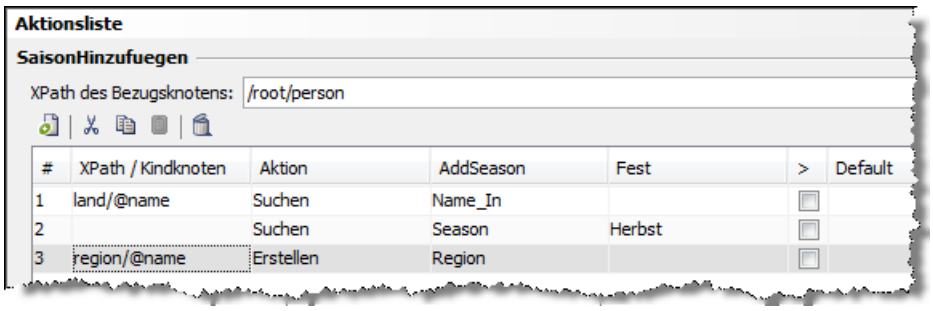
   

Name_In	Name_Out	Region	Season
Greece	Griechenland	Peloponnes	Herbst
Italy	Italien	Apulien	Winter
Italy	Italien	Rom	Herbst
Spain	Spanien	Galicien	Winter
France	Frankreich	Provence	Frühjahr
Germany	Deutschland	Bayern	Sommer

   Ergebnis: `Italy`. Dieser Wert ist in der Spalte allerdings zweimal vorhanden.
2. Prüfe zusätzlich in den beiden Zeilen, in denen der Wert `Italy` auftaucht, ob die Spalte `Season` den Wert `Herbst` hat.
   Dies trifft für Zeile 3 zu.

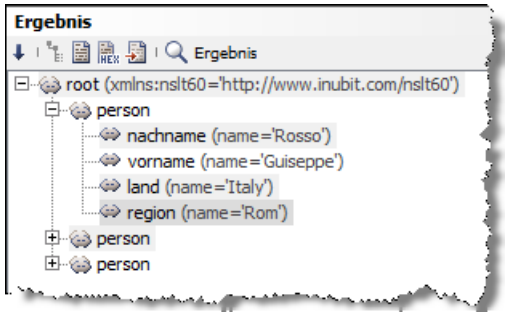
Der aktuelle Wert der Spalte `Season` in der Zeile 3 ist `Rom`, deswegen wird in der Ausgangsnachricht der Eintrag `province` mit dem Wert `Rom` erstellt.

In der Complex Lookup Table sehen diese Angaben so aus:



Ausgangsnachricht

Das Ergebnis ist folgendes:

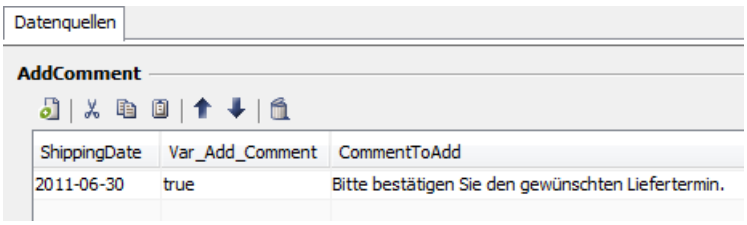


1.5 Beispiel: Variablenwert als zusätzliches Suchkriterium

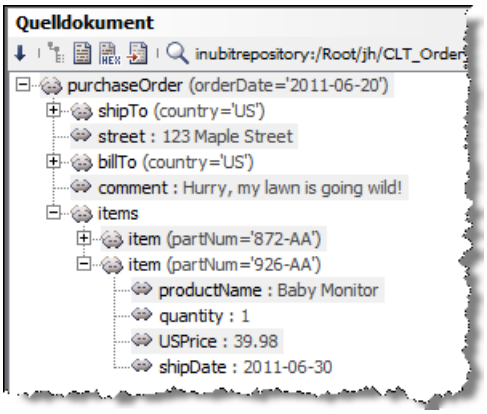
Sie können die Werte von Workflow-Variablen (Systemvariablen und selbst definierte Variablen) als zusätzliche Suchbedingung nutzen und damit die Ergebnismenge weiter einschränken.

Voraussetzungen

- Sie haben für das Variablen-Mapping an der Complex Lookup Table die Workflow-Variable `VarAddComment` wie folgt definiert:
  - Quelle: Statischer Wert, Typ: Text, Wert: true
  - Ziel: Variable `VarAddComment`, Typ: `xs:string`
- Sie haben eine Datenquelle mit folgenden Einträgen angelegt:



- Sie haben eine Eingangsnachricht mit folgender Struktur erstellt und im Bereich „XML-Quelldatei“ eingelesen:



- Sie haben eine neue Verarbeitungsvorschrift erstellt und die oben beschriebene Datenquelle aktiviert.
- Sie haben die Checkboxen „Aktiv“ und „Feste Werte als zusätzliches Suchkriterium benutzen“ aktiviert.

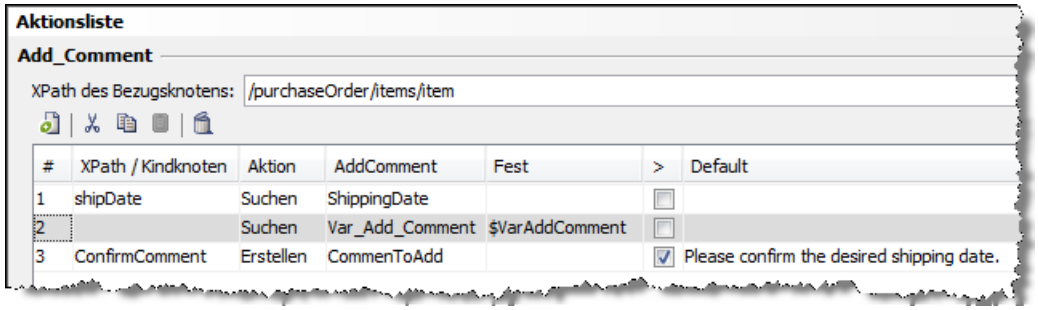
Verarbeitungsvorschrift

Erstellen Sie folgende Aktionen:

1. Suche in der Eingangsnachricht nach einem Wert für den Knoten `shipDate`, das dem ersten Wert für `ShippingDate` in der Datenquelle (`2011-06-30`) entspricht. Dazu gehen Sie wie folgt vor:
   a. Tragen Sie den folgenden XPath in das Feld „XPath des Bezugsknotens“ ein: `/purchaseOrder/items/item`
   b. Tragen Sie `shipDate` in die erste Zeile der Spalte „XPath/ Kindknoten“ ein.
   c. Wählen Sie als Aktion „Suchen“.
   d. Wählen Sie in der Spalte „AddComment“ (Name der Datenquelle) den Eintrag „ShippingDate“ (Name der ersten Spalte der Datenquelle) aus.
2. Suche in der Datenquelle nach der Spalte `Var_Add_Comment` und prüfe, ob der in der aktuellen Zeile eingetragene Wert mit dem Wert der vom Vorgängermodul übergebenen Variablen `VarAddComment` übereinstimmt. Dazu gehen Sie wie folgt vor:
   a. Wählen Sie in der zweiten Zeile als Aktion „Suchen“ aus.
   b. Wählen Sie in der zweiten Zeile der Spalte „AddComment“ den Eintrag „Var_Add_Comment“ aus.
   c. Tragen Sie in der Spalte „Fest“ den Namen der Variablen mit einem vorangestellten $-Zeichen ein: `$VarAddComment`
3. Sind beide Bedingungen erfüllt, erstelle unter dem angegebenen Bezugsknoten der Eingangsnachricht das Element „ConfirmComment“ mit dem Inhalt der Spalte „CommentToAdd“ der aktuellen Zeile der Datenquelle. Dazu gehen Sie wie folgt vor:

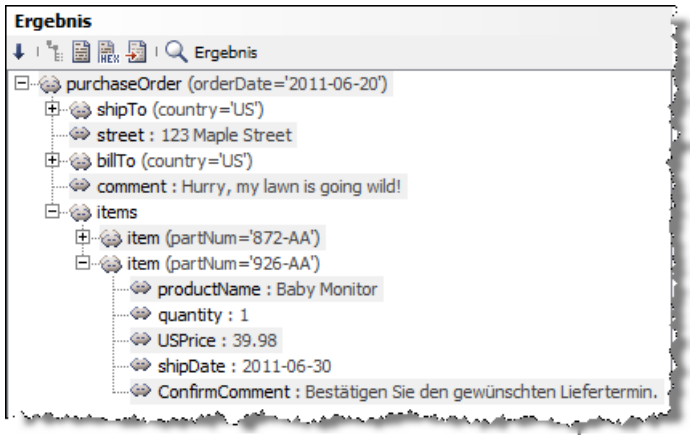
a. Tragen Sie in der Spalte „XPath/Kindknoten“ in die dritte Zeile `ConfirmComment` ein. Dieser Knoten soll unter dem Bezugsknoten neu angelegt werden.
b. Wählen Sie als Aktion „Erstellen“.
c. Wählen Sie in der Spalte „AddComment“ die Spalte „CommentToAdd“ der Datenquelle aus. Damit weisen Sie dem neuen Knoten den in der Datenquelle in der aktuellen Zeile eingetragenen Wert zu: „Bestätigen Sie das gewünschte Lieferdatum.“
d. Optional können Sie in der Spalte „Default“ einen Standardwert angeben, der dem neuen Knoten zugewiesen wird, wenn eine der Bedingungen nicht zutrifft.
e. Speichern Sie die Änderungen.

In der Complex Lookup Table sehen diese Aktionen so aus:


Aktionsliste
Add_Comment
XPath des Bezugsknotens: /purchaseOrder/items/item
XPath / Kindknoten Aktion AddComment Fest > Default
1 shipDate Suchen ShippingDate
2 Suchen Var_Add_Comment $VarAddComment
3 ConfirmComment Erstellen CommenToAdd Please confirm the desired shipping date.

Ausgangsnachricht

Das Ergebnis ist folgendes:

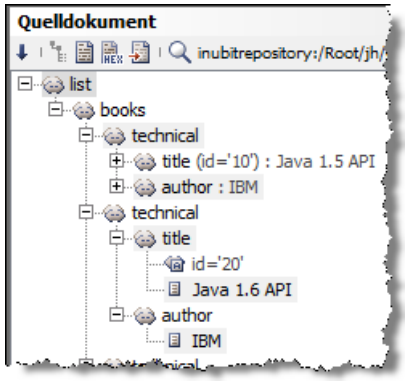


1.6 Beispiel: XPath-Ausdruck als zusätzliches Suchkriterium

Um die Suche einzugrenzen, können Sie einen XPath-Ausdruck, der sich auf Werte in Eingangsnachrichten bezieht statt auf XML- oder Datenbanktabellen, als zusätzliches Suchkriterium verwenden.

Eingangsnachricht

Die Eingangsnachricht hat folgende Struktur:

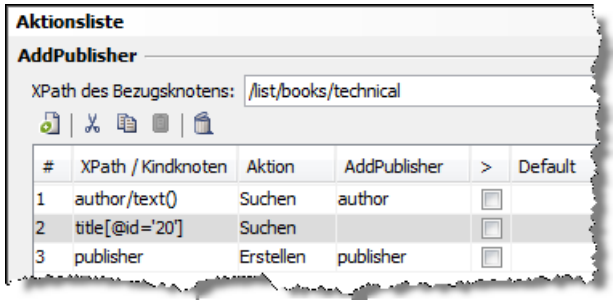


Verarbeitungsvorschrift

Die folgenden Aktionen werden erstellt:

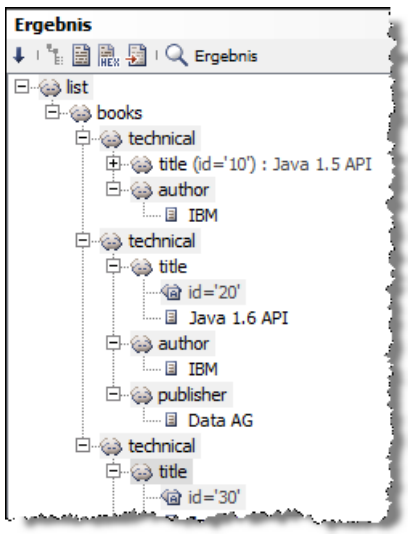
1. Suche in der Datenquelle `books` in der Spalte `author` nach dem Wert des ersten Elements `author`.
   In der Datenquelle ist dem ersten `author`-Element der Wert `IBM` zugeordnet.
2. Suche in der Eingangsnachricht nach dem Wert des Elements `title`, das gleichzeitig ein Attribut `id` mit dem Wert `20` hat.
   Dies trifft für das Buch „Java 1.6 API“ zu.
3. Erstelle unter genau diesem `title`-Element ein Element `Publisher` und gebe diesem neuen Element den Wert, welcher in der Datenquelle `books` dem in Schritt 1 gefundenen Wert (IBM) in der Spalte `publisher`. zugeordnet ist. Das ist der Wert `DATA AG`.

In der Complex Lookup Table sehen diese Aktionen so aus:



Ausgangsnachricht

Das Ergebnis ist folgendes:



1.7 Dialogbeschreibungen

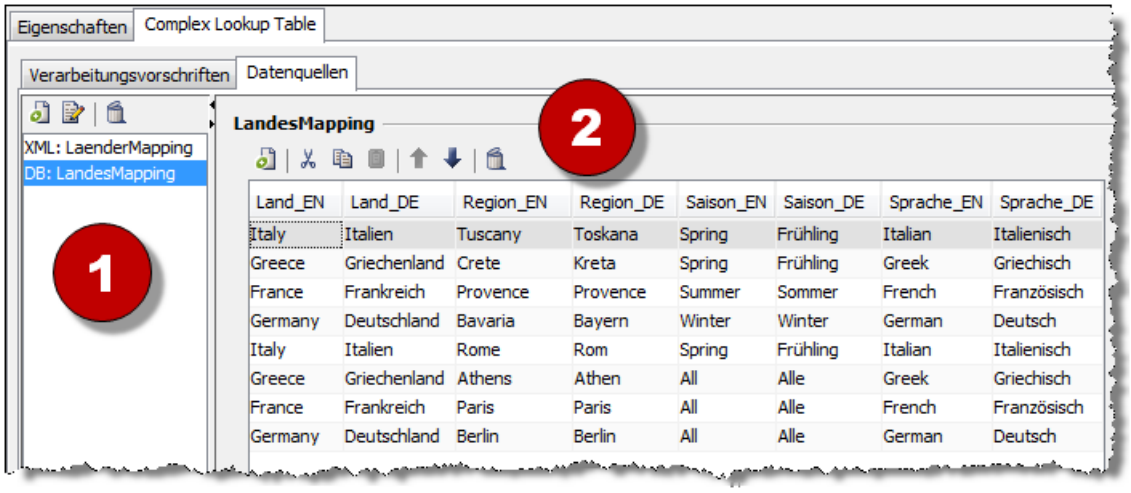
Dieser Abschnitt erläutert die folgenden Themen:

- *Register „Datenquellen“, S. 28*
- *Register „Verarbeitungsvorschriften“, S. 31*

1.7.1 Register „Datenquellen“

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Datenquelle hinzufügen“ (1), S. 29*
- *Dialog „Datenquelle hinzufügen“ (2), S. 30*


Eigenschaften
Complex Lookup Table
Verarbeitungsvorschriften
Datenquellen
XML: LaenderMapping
DB: LandesMapping
1
2
LandesMapping
Land_EN | Land_DE | Region_EN | Region_DE | Saison_EN | Saison_DE | Sprache_EN | Sprache_DE
Italy | Italien | Tuscany | Toskana | Spring | Frühling | Italian | Italienisch
Greece | Griechenland | Crete | Kreta | Spring | Frühling | Greek | Griechisch
France | Frankreich | Provence | Provence | Summer | Sommer | French | Französisch
Germany | Deutschland | Bavaria | Bayern | Winter | Winter | German | Deutsch
Italy | Italien | Rome | Rom | Spring | Frühling | Italian | Italienisch
Greece | Griechenland | Athens | Athen | All | Alle | Greek | Griechisch
France | Frankreich | Paris | Paris | All | Alle | French | Französisch
Germany | Deutschland | Berlin | Berlin | All | Alle | German | Deutsch

Im Register „Datenquellen“ verwalten Sie statische XML-Datenquellen und dynamische DB-Datenquellen. Das Register besteht aus folgenden Bereichen:

1. **Liste aller Datenquellen**
   Zeigt eine Liste aller bereits vorhandenen Datenquellen an.
   → Siehe *Datenquelle erstellen (Workbench/Process Engine: Modul-Guide, Kap. 1, S. 17)*
2. **Datenansicht**
   Zeigt die Spaltenstruktur der im Bereich „Liste der Datenquellen“ ausgewählten Datenquelle an.
   Über das Kontextmenü können Sie die Spaltenstruktur einer XML-Datenquelle ändern oder die Tabelleneigenschaften einer DB-Datenquelle anpassen.
   Das Hinzufügen, Ändern und Verschieben von Werten sowie das Importieren von Tabellendaten ist hier nur für XML-Datenquellen möglich. Die Werte einer DB-Datenquelle ändern Sie direkt in der Datenbanktabelle.

1.7.1.1 Dialog „Datenquelle hinzufügen“ (1)

In diesem Dialog wählen Sie den Typ der neuen Datenquelle aus und benennen diese.

Typ

- **XML-Datenquelle**
  XML-Datenquellen erstellen Sie direkt im Complex Lookup-Modul. Alternativ können Sie auch vorhandene XML-Datenquellen importieren.
  → Siehe *Verarbeitungsvorschriften anlegen (Workbench/Process Engine: Modul-Guide, Kap. 1.2, S. 17).*
- **DB-Datenquelle**

Eine DB-Datenquelle referenziert eine externe Datenbanktabelle.

Informationen der Datenquelle

- **Name**
  Name der Datenquelle. Der Name kann ein anderer sein als der Name, der in der Datenbank verwendet wird.
- **Beschreibung**: Optional.

1.7.1.2 Dialog „Datenquelle hinzufügen“ (2)

(Nur bei Typ der Datenquelle = DB-Datenquelle)

In diesem Dialog geben Sie die Verbindungsdaten zur Datenbank ein.

Datenquelle

- **Datenbank**
  Wählen Sie einen Datenbank- oder Business Object Connector aus, dessen aktuelle Verbindungsparameter kopiert und zum Herstellen der Datenbankverbindung genutzt werden sollen. Sie können diese Werte nachträglich ändern.

Datenbankverbindung

- **Voreinstellung**
  Zur Auswahl eines Datenbanktyps. Auf Basis der Auswahl werden in den folgenden Feldern eine Datenbank-Muster-URL und der korrekte Datenbank-Treiber gesetzt.
  Wählen Sie die Option „Custom“, um keine Vorgaben zu setzen.
- **Datenbank-URL**
  Wenn im Feld „Voreinstellung“ eine Auswahl getroffen wurde, dann wird eine korrekte Muster-URL zum Ansprechen der Datenbank angezeigt.
  Um eine andere Datenbank anzusprechen, als unter „Voreinstellung“ angegeben, ignorieren Sie die Voreinstellung und geben die korrekte URL Ihrer Datenbank ein.
- **Datenbanktreiber-Klasse**
  Name der Datenbanktreiber-Klasse. Wird automatisch korrekt gesetzt, wenn unter „Voreinstellung“ eine Datenbank ausgewählt wurde.

Der zur gewählten Datenbank passende Datenbanktreiber muss installiert sein! Siehe *Treiber installieren (Process Engine: Administrator- und Entwickler-Guide, Kap. 3.4, S. 51).*

- **Benutzer**
  Gültiger Benutzername für die ausgewählte Datenbank.

- **Passwort**
  Passwort, mit dem sich der Benutzer gegenüber der Datenbank authentifiziert.
- **Textwerte aus der Datenbank trimmen**
  Löscht bei Textwerten führende und nachfolgende Leerstellen.
- **Spezielle Kodierung setzen**
  Wenn markiert, dann können Sie einen Zeichensatz auswählen oder einen beliebigen Zeichensatz eingeben.
- **Connection Pooling**
  Öffnet den *Dialog „Datenbank Connection Pooling“ (Workbench/ Process Engine: Systemkonnektor-Guide, Kap. 6.5.5, S. 87)*, in dem Sie die Wiederverwendung physischer Datenbankverbindungen konfigurieren, um den zeitaufwändigen Auf- und Abbau von Datenbankverbindung zu optimieren.
- **Spezielle Verbindungsparameter**
  Zum Erstellen von Parametern, z. B. Timeouts und Anzahl offener Verbindungen, mit denen Sie die Datenbankverbindung abhängig von der jeweils verwendeten Datenbank konfigurieren. Die Parameter definieren Sie als Name-Wert-Paare.

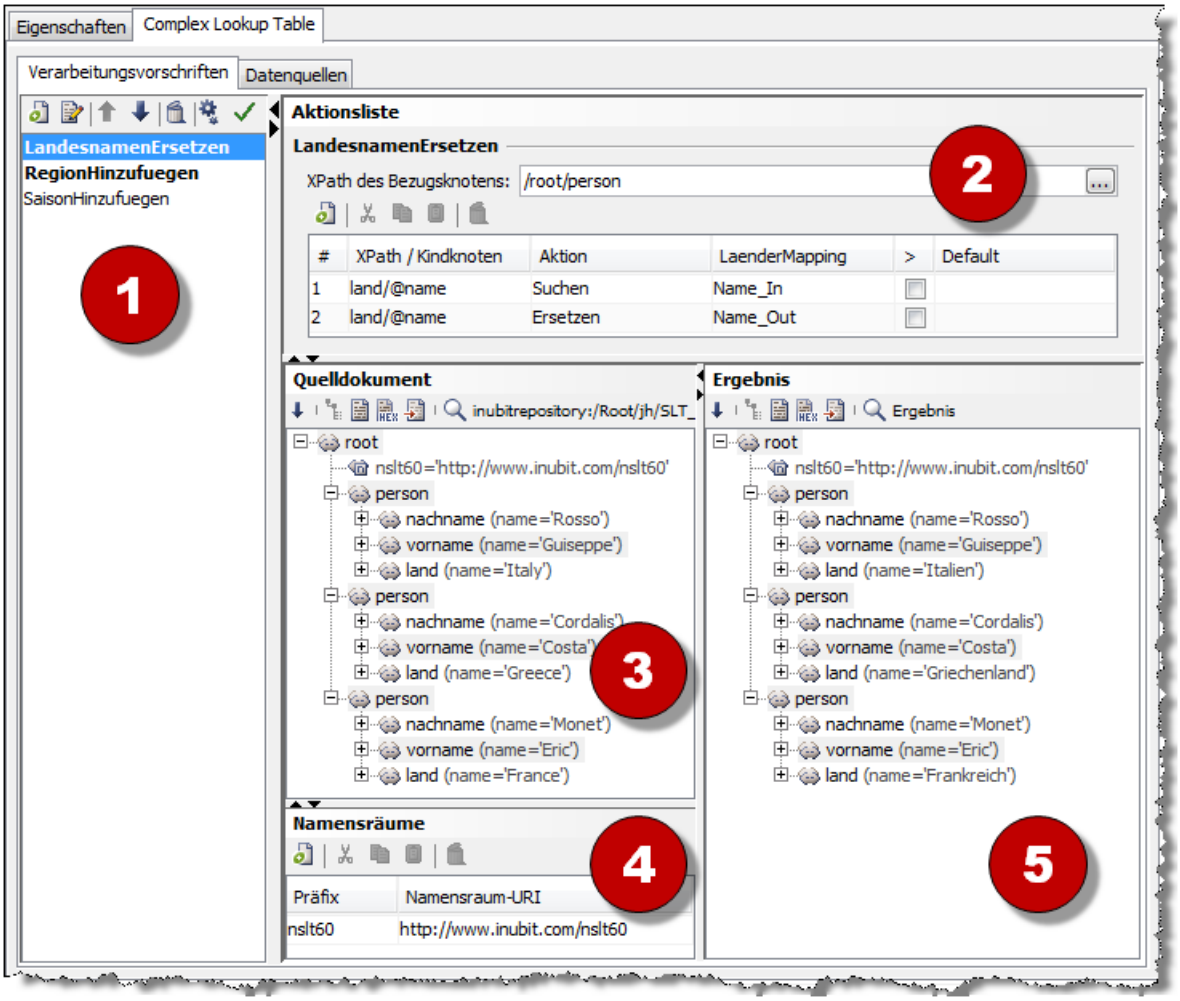
Verbindungstest

- **Verbindung testen**
  Zum Testen, ob die Verbindung mit Ihren Angaben erfolgreich aufgebaut werden kann.

1.7.2 Register „Verarbeitungsvorschriften“

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Eigenschaften der Verarbeitungsvorschrift“, S. 33*
- *Bereich „Aktionsliste“, S. 34*



Im Register „Verarbeitungsvorschriften“ erstellen und konfigurieren Sie Verarbeitungsvorschriften. Das Register besteht aus folgenden Bereichen:

1. **Liste aller Verarbeitungsvorschriften**

   Zeigt eine Liste aller bereits vorhandenen Verarbeitungsvorschriften an.

   Sie können Verarbeitungsvorschriften über das Kontextmenü anlegen, bearbeiten und löschen.

   → Siehe *Verarbeitungsvorschriften anlegen (Workbench/Process Engine: Modul-Guide, Kap. 1.2, S. 17).*

2. **Aktionsliste**

   Zur Konfiguration der ausgewählten Verarbeitungsvorschrift und zum Anlegen der einzelnen Aktionen.

   → Siehe *Bereich „Aktionsliste“ (Workbench/Process Engine: Modul-Guide, Kap. 1.7.2.2, S. 34).*

3. **Quelldokument**

   Zum Öffnen und Anzeigen einer Beispiel-Eingangsnachricht.

Um die XPath-Angaben im Bereich „Aktionslisten“ zu erstellen, wird die Eingangsnachricht automatisch im XPath-Assistenten geöffnet.

4. **Namensräume**
   In diesem Bereich werden die Namensräume angezeigt, die in der Eingangsnachricht definiert sind. Sie können Namensraumdefinitionen hinzufügen, ändern und löschen.
5. **Ergebnis**
   In diesem Bereich wird das Ergebnis angezeigt, wenn Sie die Verarbeitungsvorschriften getestet haben.
   → Siehe *Verarbeitungsvorschrift testen (Workbench/Process Engine: Modul-Guide, Kap. 7., S. 20).*

1.7.2.1 Dialog „Eigenschaften der Verarbeitungsvorschrift“

Verarbeitungsvorschrift

- **Name**
  Eindeutiger Name der Verarbeitungsvorschrift.

Eigenschaften der Verarbeitungsvorschrift

- **Aktiv**
  Legt fest, ob eine Verarbeitungsvorschrift ausgeführt werden soll. Nur aktive Verarbeitungsvorschriften werden ausgeführt.
- **Feste Werte als zusätzliches Suchkriterium benutzen**
  Markieren Sie diese Option, wenn der Wert in einer Spalte nicht ausreicht, um eine Zeile eindeutig zu identifizieren, oder wenn Sie ausschließlich nach Werten aus einer bestimmten Zeile suchen wollen.
  Wenn diese Option aktiviert ist, können Sie auch Workflow-Variablen als zusätzliches Suchkriterium nutzen.
  → Siehe
  - *Beispiel: Feste Werte als zusätzliches Suchkriterium (Workbench/Process Engine: Modul-Guide, Kap. 1.4, S. 22)*
  - *Beispiel: Variablenwert als zusätzliches Suchkriterium (Workbench/Process Engine: Modul-Guide, Kap. 1.5, S. 24)*
- **Groß-/Kleinschreibung beachten**
  Wenn markiert, dann wird bei der Ersetzung der Werte die Groß- und Kleinschreibung beachtet.
- **Fehler anzeigen, wenn kein Wert gefunden wurde**:
  Wenn markiert, dann wird ein Fehler gemeldet, falls zwar ein Element zur Abbildungsvorschrift vorhanden, aber ungleich dem Wert in der Datenquelle ist.
  - **Fehler ignorieren, wenn Quellknoten nicht gefunden wurde**

Wenn markiert, dann wird kein Fehler gemeldet, obwohl zu einer Abbildungsvorschrift kein passendes Element gefunden wird.

- **Bei der Datenbankabfrage leere Zeichenkette zu NULL konvertieren**
  Wenn markiert, dann werden leere Zeichenketten durch NULL ersetzt. Wählen Sie diese, wenn bei der DB-Abfrage undefinierte, leere Werte als NULL-Werte ausgegeben werden sollen.

Verwendete Datenquellen

Die Tabelle listet alle definierten Datenquellen mit Name, Typ und Kommentar auf. Markieren Sie die Datenquellen, die Verarbeitungsvorschrift verwenden soll.

1.7.2.2 Bereich „Aktionsliste“

- **XPath des Bezugsknotens**
  Pfad zu dem Element, welches als Referenz für alle Aktionen dient. Alle Aktionen müssen sich auf Elemente beziehen, die hierarchisch unterhalb des Bezugsknotens liegen.
- **Tabelle**
  - **#** (Erste Spalte)
    Laufende Nummer der Zeile.
  - **XPath/Kindknoten**
    Pfad zu einem Element der XML-Eingangsnachricht. Markieren Sie dazu das Element in Bereich „XML-Quelldatei“ und ziehen Sie es in die gewünschte Zeile.
  - **Aktion**
    Mögliche Aktionen sind „Suchen“, „Ersetzen“ und „Erstellen“.
    → Siehe *Aktionen definieren (Workbench/Process Engine: Modul-Guide, Kap. 6., S. 19).*
  - **<Name der Datenquelle>**
    Name der Datenquellen, die im *Dialog „Eigenschaften der Verarbeitungsvorschrift“ (Workbench/Process Engine: Modul-Guide, Kap. 1.7.2.1, S. 33)* in der Tabelle „Verwendete Datenquellen“ aktiviert sind.
    In den Zellen können die Tabellenspalten aus DropDown-Listen ausgewählt werden.
  - **Fest**
    Tragen Sie einen festen Wert ein oder referenzieren Sie eine Workflow-Variable.
    → Siehe

- *Beispiel: Feste Werte als zusätzliches Suchkriterium (Workbench/Process Engine: Modul-Guide, Kap. 1.4, S. 22)*
- *Beispiel: Variablenwert als zusätzliches Suchkriterium (Workbench/Process Engine: Modul-Guide, Kap. 1.5, S. 24)*

Diese Spalte wird nur angezeigt, wenn Sie im *Dialog „Eigenschaften der Verarbeitungsvorschrift“ (Workbench/ Process Engine: Modul-Guide, Kap. 1.7.2.1, S. 33)* die Option „Feste Werte als zusätzliches Suchkriterium benutzen“ aktiviert haben.

- **> (Operator)**
  Wenn markiert und wenn als Aktion „Ersetzen“ oder „Erstellen“ ausgewählt ist, dann können Sie in der Spalte „Default“ einen Standardwert eingeben.
- **Default**
  Standardwert. Das XML-Element in derselben Zeile wird immer dann durch den Default ersetzt bzw. damit erstellt, wenn kein Wert gefunden wurde, der die angegebenen Bedingungen erfüllt.

Wenn die Option „Fehler anzeigen, wenn kein Wert gefunden wurde“ aktiviert ist, dann endet die Verarbeitung, wenn eine der angegebenen Bedingungen nicht erfüllt wurde. In diesem Fall wird der Standardwert ignoriert.

Dieser Abschnitt erläutert die folgenden Themen:

- *PDF/A-1b erstellen und Fonts ändern, S. 37*
- *UNIX/Linux: SVG-Grafiken erzeugen, S. 40*
- *Dialog „FO Converter Eigenschaften“, S. 40*

Verwendung

Mit dem FO Converter konvertieren Sie XSL-FO-formatierte Eingangsnachrichten in eines der folgenden Ausgabeformate:

- **PDF**
- **PDF/A-1b**
  → Siehe *PDF/A-1b erstellen und Fonts ändern (Workbench/Process Engine: Modul-Guide, Kap. 2.1, S. 37)*.
- **PDF/X-3:2003**
- **PCL** (Printer Control Language)
- **PostScript**
- **RTF** (Rich Text Format)
- **SVG** (Scalable Vector Graphic)
  → Siehe *UNIX/Linux: SVG-Grafiken erzeugen (Workbench/Process Engine: Modul-Guide, Kap. 2.2, S. 40)*.
- **TXT**
- **WML** (Word Markup Language)

→ Siehe auch *Ausgabeformat (Workbench/Process Engine: Modul-Guide, Kap. 2, S. 40)*.

Zum Erstellen einer XSL-FO-formatierten Eingangsnachricht können Sie z. B. den XSLT Converter einsetzen, siehe *XSLT Converter (Data Converter) (Workbench/Process Engine: Modul-Guide, Kap. 5, S. 133)*.

2.1 PDF/A-1b erstellen und Fonts ändern

Voraussetzungen

- Sie benötigen eine Font-Metrics-Datei. Diese Datei enthält allgemeine und zeichensatzspezifische Informationen für die zu benutzenden Fonts.
- Verwenden Sie keine PDF-Verschlüsselung. Diese ist für PDF/A nicht zulässig.
- Verwenden Sie keine CMYK-Grafiken ohne ICC Color-Profile.

Der FO-Prozessor unterstützt derzeit nur den sRGB-Farbraum zuverlässig und das Mischen von Farbräumen ist nicht zulässig.

FOP bettet standardmäßig ein Standard-sRGB ICC-Profil ein (sRGB IEC61966-2.1). Dieses Profil wird verwendet, wenn keine anderen Angaben in der Konfigurationsdatei vorhanden sind.

- Verwenden Sie in SVG-Grafiken nur RGB-Farben.
- Setzen Sie bei `fo:external-graphic` keine EPS-Grafiken ein. Eingebettete EPS-Grafiken sind bei PDF/A nicht zulässig.
- Sie dürfen keinen expliziten Filter für Metadaten-Objekte angeben. Metadaten müssen im Klartext eingebettet werden, damit nicht-PDF-fähige Applikationen die XMP-Metadaten extrahieren können.
- In den Eingangsnachrichten muss im `root`-Element das Attribut `font-family` angegeben sein:

```
<?xml version="1.0" encoding="UTF-8"?>
 <fo:root
   xmlns:fo="http://www.w3.org/1999/XSL/Format"
   font-family="Arial">
[…]
</fo:root>
```

- Stellen Sie sicher, dass alle(!) Fonts eingebettet sind. Wenn Sie base-14-Fonts (z. B. Helvetica) verwenden, müssen Sie diese lizenzieren und genauso einbetten wie andere Fonts.
- Die PDF-Version 1.4 wird erzwungen, wenn die PDF/A-1b-Ausgabe aktiviert ist.

So gehen Sie vor

1. Erzeugen Sie eine Konfigurationsdatei für den FO-Prozessor, z. B. `fop.xconf` mit folgendem Inhalt:

```
<?xml version="1.0"?>
<fop version="1.0">
 <!-- Base URL for resolving relative URLs -->
 <base>/home/vk/a-tests/fop/pdfa-1b</base>
 <!-- Source resolution in dpi (dots/pixels per
 inch) for determining the size of pixels in SVG
 and bitmap images, default: 72dpi -->
 <source-resolution>72</source-resolution>
 <!-- Target resolution in dpi (dots/pixels per
 inch) for specifying the target resolution for
 generated bitmaps, default: 72dpi -->
 <target-resolution>300</target-resolution>
 <!-- Default page-height and page-width, in case
 value is specified as auto -->
 <default-page-settings height="29.7cm"
 width="21cm"/>
 <!-- Information for specific renderers -->
```

```
 <!-- Uses renderer mime type for renderers -->
 <renderers/>
</fop>
```

2. Geben Sie im Element `<base>` das Basis-Verzeichnis für Fonts an, z. B. `<base>c:\fop\fonts</base>`.
3. Kopieren Sie den einzubettenden Font und seine Font-Metrics in das Basis-Verzeichnis.
4. Fügen Sie der FOP-Konfigurationsdatei die Definition des einzubettenden Fonts hinzu, z. B.:

```
<renderers>
   <renderer mime="application/pdf">
     <fonts>
       <font
         metrics-url="arial.xml"
         kerning="yes"
         embed-url="arial.ttf">
       <font-triplet
         name="Arial"
         style="normal"
         weight="normal"/>
       </font>
     </fonts>
 </renderer>
</renderers>
```

5. Erstellen Sie einen FO Converter:
   a. Legen Sie folgende Einstellungen fest:
      - XSL-FO-Prozessor: Apache FOP
      - Ausgabeformat: PDF/A-1B.
   b. Klicken Sie auf „Fertig stellen“.
6. Um die eben erstellte Konfigurationsdatei zu hinterlegen, fügen Sie folgende Moduleigenschaft hinzu:
   - Name: `ConfigFile`
   - Datentyp: `Text (einzeilig)`
   - Wert: Pfad zu der FOP-Konfigurationsdatei

   → Siehe *Moduleigenschaften hinzufügen (Workbench: Benutzer-Guide, Kap. 3.7, S. 123).*
7. Speichern Sie Ihre Änderungen.

2.2 UNIX/Linux: SVG-Grafiken erzeugen

Wenn kein X-Server installiert ist, müssen Sie Dateien anpassen, um SVG-Grafiken erzeugen zu können.

So gehen Sie vor

1. Öffnen Sie eine der folgenden Dateien, abhängig vom verwendeten Applikationsserver:
   - JBoss: `<iS-installdir>server/JBoss/bin/run.sh`
   - Tomcat: `<iS-installdir>server/Tomcat/bin/setenv.sh` bzw. `.bat`
2. Fügen Sie unter der Variablen „JAVA_OPTS“ folgenden Parameter hinzu: `-Djava.awt.headless=true`
3. Speichern Sie die Änderung.
4. Starten Sie den Applikationsserver neu.

2.3 Dialog „FO Converter Eigenschaften“

In diesem Dialog haben Sie folgende Optionen:

- **XSL-FO Prozessor**
  Die Prozessoren unterscheiden sich in ihrem Funktionsumfang und den Ausgabeformaten, die sie unterstützen:
  - **Apache FOP**
    Unterstützt PDF/A-1b als Archivformat und bietet eine bessere Unterstützung des FOP-Standards.
  - **Apache FOP 0.20.5**
    Standardprozessor in iS-Versionen < 5.1.

Diese Version wird nicht mehr weiterentwickelt!

  - **XMLmind XSL-FO**
    Unterstützt Ausgabe als RTF und WordML.
- **Ausgabeformat**
  Abhängig vom gewählten XSL-FO-Prozessor kann eines der folgenden Ausgabeformate erzeugt werden:
  - **PDF**
    Portable Document Format der Firma Adobe, kann mit dem kostenlosen Acrobat Reader angezeigt werden.
  - **PDF/A-1b**

Standard für die Langzeitarchivierung elektronischer Dokumente. In einem PDF/A-1b-Dokument müssen in jedem Fall folgende Informationen enthalten sein:

- Alle benutzten Fonts (Untergruppen sind möglich)
- Alle Bilder
- Metadaten im XMP-Format

→ Siehe *PDF/A-1b erstellen und Fonts ändern (Workbench/ Process Engine: Modul-Guide, Kap. 2.1, S. 37).*

- **PDF/X-3:2003**

  ISO-Standard, der die Übermittlung digitaler PDF-Druckvorlagen regelt. Wenn verwendet, müssen die folgenden Daten eingebettet sein:

  - Schriften
  - TrimBox
  - Bilddaten

  Nicht erlaubt sind:

  - OPI-Kommentare
  - Transferkurven
  - Kommentare und Formularfelder
  - Verschlüsselung
  - Transparenzen

- **PCL** (Printer Control Language)

  Ein von der Firma Hewlett and Packard definiertes Format zur Steuerung von Druckern. Ausgaben in PCL können direkt an einen Drucker geschickt werden.

- **PostScript**

  Von der Firma ADOBE entwickelter Standard zur Layout-Beschreibung einer gedruckten Seite. Eine PostScript formatierte Seite kann von speziellen Anzeigenprogrammen angezeigt werden oder direkt an einen Drucker geschickt werden, wenn dieser das PostScript Format unterstützt.

- **RTF** (Rich Text Format)

  Dateiformat für den Austausch formatierter Texte zwischen verschiedenen Textverarbeitungsprogrammen. Größe, Farbe und Art der Schriften eines Textes bleiben auch beim Austausch zwischen Programmen verschiedener Hersteller erhalten.

- **SVG** (Scalable Vector Graphic)

  XML-basiertes Format für Vektorgrafiken.

  Wenn eine SVG-Datei in eine HTML-Seite integriert wird, dann muss der Browser über ein Plug-in verfügen, das dieses Format darstellen kann.

  → Siehe *UNIX/Linux: SVG-Grafiken erzeugen (Workbench/ Process Engine: Modul-Guide, Kap. 2.2, S. 40).*

- **TXT**
  Textdatei ohne Layout-Informationen.
- **WML** (Word Markup Language)
  XML-Format, von MS Word ab Version 2003 verwendet.

- **Arbeitsverzeichnis des FO Converters**
  Absoluter Pfad zu Multimediadateien (Bilder, Videos etc.), welche in das Ausgabeformat eingebunden werden sollen.

3 Lookup Table (Data Converter)

Dieser Abschnitt erläutert die folgenden Themen:

- *Funktionsprinzip, S. 43*
- *Datenquelle anlegen, S. 45*
- *Datenbanktabelle referenzieren, Tabellenname und Spaltennamen setzen, S. 49*
- *Tabellen und Spalten mit dem DB-Manager definieren, S. 50*
- *Ersetzungsvorschriften definieren, S. 51*
- *Dialogbeschreibungen, S. 52*

Verwendung

Mit einer Lookup Table ändern Sie tabellengesteuert Werte in Nachrichten.

3.1 Funktionsprinzip

Ersetzungsvorschriften

Eine Lookup Table kann eine Vorschrift ausführen, die aus mehreren Ersetzungsvorschriften besteht. Die zu ersetzenden Werte und die Ersatzwerte werden in Tabellen (Datenquellen) aufeinander abgebildet. Als Datenquellen können Sie statische XML-Tabellen und dynamische Datenbanktabellen verwenden:

- **XML-Datenquelle**

  Geeignet, wenn nur wenige Werte geändert werden sollen.

  XML-Datenquellen werden direkt in der Lookup Table angelegt und mit Wertepaaren gefüllt.

  → Siehe *XML-Datenquelle definieren (Workbench/Process Engine: Modul-Guide, Kap. 3.2.1, S. 46).*

- **DB-Datenquelle**

  Geeignet, um viele Konvertierungsschritte durchzuführen, weil die Tabelle nur einmal geladen und dann im Speicher gehalten wird.

  → Siehe *DB-Datenquelle definieren (Workbench/Process Engine: Modul-Guide, Kap. 3.2.5, S. 48).*

Stapelverarbeitung

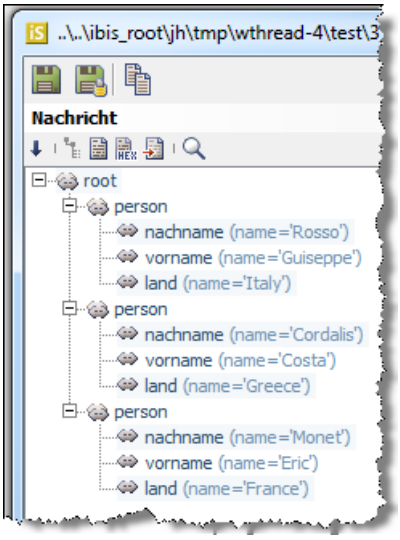
Die Ersetzungsvorschriften werden in der Reihenfolge ihres Vorkommens abgearbeitet. Jede Ersetzungsvorschrift wird genau einmal angewendet.

Die Lookup Table ruft die erste Ersetzungsvorschrift auf und verarbeitet damit die Eingangsnachricht. Dann wird die nächste Ersetzungsvorschrift auf die bereits veränderte Nachricht angewendet. Die Eingangsnachricht verändert sich schrittweise mit jeder Ersetzungsvorschrift.

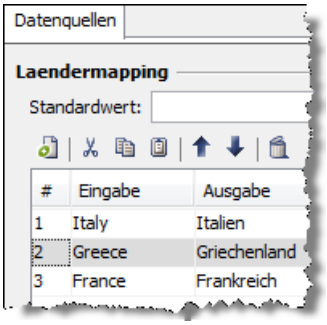
Beispiel

Im folgenden Beispiel werden englische Ländernamen durch deutsche ersetzt.

- Die Eingangsnachricht hat folgende Struktur:

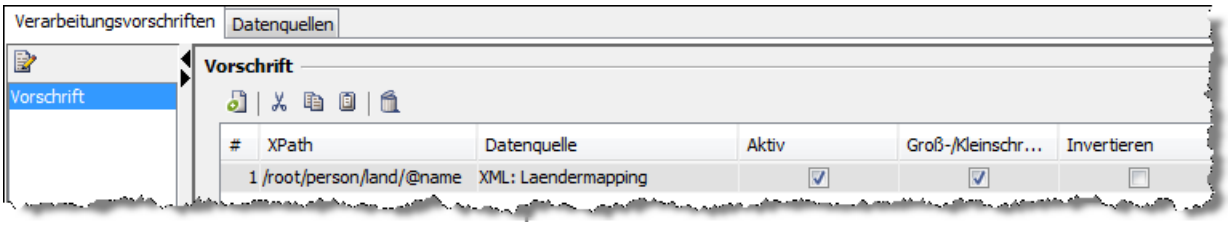


- Die Ersetzungsvorschrift lautet:
  a. Suche in der Eingangsnachricht nach dem ersten Vorkommen des Elements `/root/person/land` mit dem Wert `Italy`.
  b. Ersetze `Italy` mit dem Wert, der in der XML-Datenquelle `Laendermapping` in der Spalte `Ausgabe` in derselben Zeile wie `Italy` angegeben ist, hier ist das `Italien`:

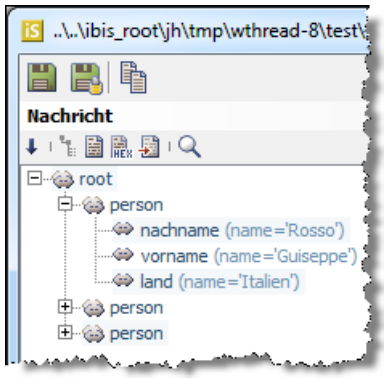


  c. Verfahre analog mit allen weiteren Werten der `land`-Elemente, die in der XML-Tabelle angegeben sind.

  In der Lookup Table sieht die Ersetzungsvorschrift so aus:



- Die **Ausgangsnachricht** sieht so aus:



3.2 Datenquelle anlegen

Dieser Abschnitt erläutert die folgenden Themen:

- *XML-Datenquelle definieren, S. 46*
- *Daten in eine XML-Datenquelle importieren, S. 46*
- *Daten manuell zu einer XML-Datenquelle hinzufügen, S. 47*
- *Daten einer XML-Datenquelle bearbeiten und löschen, S. 48*
- *DB-Datenquelle definieren, S. 48*

Verwendung

Eine Datenquelle enthält die Werte für die Eingabe und die Ausgabe. Eine Ersetzungsregel beschreibt, wie die Eingabewerte durch die Ausgabewerte ersetzt werden sollen. Einer XML-Datenquelle weisen Sie die Werte direkt in der Lookup Table zu, eine DB-Datenquelle verknüpfen Sie mit einer externen Datenbank.

3.2.1 XML-Datenquelle definieren

Verwendung

Eine XML-Datenquelle dient dazu, die Eingabe- und Ausgabewerte direkt in der Lookup Table zu pflegen.

Voraussetzungen

Sie haben eine Lookup Table zum Bearbeiten geöffnet.

So gehen Sie vor

1. Wechseln Sie auf das Register „Datenquellen“.
2. Öffnen Sie im linken Bereich das Kontextmenü und wählen Sie „Hinzufügen“. Der *Dialog „Datenquelle hinzufügen“ (1) (Workbench/Process Engine: Modul-Guide, Kap. 3.6.1.1, S. 53)* öffnet sich.
3. Wählen Sie die Option „XML-Datenquelle“.
4. Benennen Sie die Datenquelle.
5. Optional: Geben Sie eine detaillierte Beschreibung der Datenquelle ein.
6. Klicken Sie auf „Fertigstellen“, um die Datenquelle zu speichern.

Die neue Datenquelle erscheint in der Liste der Datenquellen. Im rechten Fensterbereich wird eine leere Tabelle mit den beiden Spalten „Eingabe“ und „Ausgabe“ angezeigt.

Nun können Sie Daten importieren oder manuell hinzufügen:

→ Siehe

- *Daten in eine XML-Datenquelle importieren (Workbench/ Process Engine: Modul-Guide, Kap. 3.2.2, S. 46)*
- *Daten manuell zu einer XML-Datenquelle hinzufügen (Workbench/Process Engine: Modul-Guide, Kap. 3.2.3, S. 47)*

3.2.2 Daten in eine XML-Datenquelle importieren

Voraussetzungen

Die XML-Tabellen müssen folgende Struktur haben:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Table>
    Row in="EingehenderWert"
    out="AusgegebenerWert"/>
    <Row in="EingehenderWert"
    out="AusgegebenerWert"/>
```

```
</Table>
```

So gehen Sie vor

1. Zeigen Sie das Register „Datenquellen“ an.
2. Öffnen Sie im rechten Bereich das Kontextmenü und wählen Sie „Tabellendaten importieren“. Ein Dateiexplorer öffnet sich.
3. Navigieren Sie zu der XML-Datei, welche die Daten enthält, und klicken Sie „Öffnen…“.

Die Daten werden geladen und angezeigt.

3.2.3 Daten manuell zu einer XML-Datenquelle hinzufügen

Voraussetzungen

- Sie haben eine Lookup Table zum Bearbeiten geöffnet.
- Sie haben eine XML-Datenquelle definiert.

So gehen Sie vor

1. Zeigen Sie das Register „Datenquellen“ an.
2. Wählen Sie die Datenquelle aus, der Sie Werte hinzufügen wollen.
3. Optional: Geben Sie einen Standardwert ein, den das Modul verwendet, wenn es für einen Wert in der Eingangsnachricht in der Datenquelle keinen Eintrag findet.
4. Öffnen Sie im rechten Bereich das Kontextmenü und wählen Sie „Hinzufügen“, um in der Tabelle eine neue Zeile einzufügen.
5. Doppelklicken Sie in der Spalte „Eingabe“ das Feld der neuen Zeile, um den Wert aus der Eingangsnachricht einzutragen.
6. Doppelklicken Sie in der Spalte „Ausgabe“ das Feld der neuen Zeile, um den Wert für die Ausgangsnachricht einzutragen.
7. Speichern Sie die Eingaben.
8. Wiederholen Sie die Schritte 3 bis 6 für alle möglichen Werte der Eingangsnachricht.

→ Für Informationen über das Bearbeiten und Löschen von Werten einer XML-Datenquellesiehe „*Daten einer XML-Datenquelle bearbeiten und löschen (Workbench/Process Engine: Modul-Guide, Kap. 3, S. 48)*“.

3.2.4 Daten einer XML-Datenquelle bearbeiten und löschen

Voraussetzungen

- Sie haben eine Lookup Table zum Bearbeiten geöffnet.
- Sie haben eine XML-Datenquelle definiert.

Wertepaar bearbeiten

So gehen Sie vor

1. Zeigen Sie das Register „Datenquellen“ an.
2. Wählen Sie die Datenquelle aus, deren Werte Sie ändern wollen.
3. Führen Sie einen der folgenden Schritte aus:
   - Doppelklicken Sie in der Spalte „Eingabe“ in das Feld des gewünschten Eintrags, um einen neuen Wert aus der Eingangsnachricht einzutragen.
   - Doppelklicken Sie in der Spalte „Ausgabe“ in das Feld der gewünschten Zeile, um einen neuen Wert für die Ausgangsnachricht einzutragen.
4. Speichern Sie die Eingaben.
5. Wiederholen Sie die Schritte 3 und 4 für alle zu ändernden Werte.

Wertepaar löschen

Wenn Sie einen Eintrag in der Datenquelle nicht mehr benötigen, öffnen Sie das Kontextmenü und wählen für die betreffende Zeile den Punkt „Löschen“.

3.2.5 DB-Datenquelle definieren

Verwendung

Eine DB-Datenquelle dient dazu, die Eingabe- und Ausgabewerte aus einer externen Datenbank auszulesen. Diese Datenbank wird außerhalb der Lookup Table gepflegt.

Voraussetzungen

Sie haben eine Lookup Table zum Bearbeiten geöffnet.

So gehen Sie vor

1. Zeigen Sie das Register „Datenquellen“ an.
2. Öffnen Sie im linken Bereich das Kontextmenü und wählen Sie „Hinzufügen“. Der *Dialog „Datenquelle hinzufügen“ (1) (Workbench/Process Engine: Modul-Guide, Kap. 3.6.1.1, S. 53)* öffnet sich.

3. Wählen Sie die Option „DB-Datenquelle“.
4. Benennen Sie die Datenquelle.
5. Tragen Sie optional eine detaillierte Beschreibung der Datenquelle ein.
6. Klicken Sie auf „Weiter“. Der *Dialog „Datenquelle hinzufügen“ (2) (Workbench/Process Engine: Modul-Guide, Kap. 3.6.1.2, S. 53)* öffnet sich.
7. Konfigurieren Sie die Datenbankeinstellungen.
8. Klicken Sie auf „Fertigstellen“, um die Datenquelle zu speichern.

Die neue Datenquelle wird im linken Bereich angezeigt. Im rechten Bereich wird eine leere Tabelle abgezeigt.

Nun referenzieren Sie die gewünschte Datenbanktabelle.

→ Siehe *Datenbanktabelle referenzieren, Tabellenname und Spaltennamen setzen (Workbench/Process Engine: Modul-Guide, Kap. 3.3, S. 49).*

Verbindungseinstellungen exportieren/importieren

Über das Kontextmenü einer Datenquelle können Sie die Verbindungseinstellungen als XML-Struktur exportieren und importieren.

3.3 Datenbanktabelle referenzieren, Tabellenname und Spaltennamen setzen

Voraussetzungen

- Sie haben eine Lookup Table zum Bearbeiten geöffnet.
- Sie haben eine DB-Datenquelle definiert.
- Sie haben eine Datenbank mit mindestens einer Tabelle und zwei Spalten angelegt.
  - Sie haben festgelegt, welche Spalte die Eingabe- bzw. Ausgabewerte enthält.
  - Die Spalte mit den Eingabewerten muss die in der Eingangsnachricht erwarteten Werte enthalten.
  - Die Spalte mit den Ausgabewerten muss die Werte enthalten, mit denen die Eingangswerte ersetzt werden sollen.

So gehen Sie vor

1. Zeigen Sie das Register „Datenquellen“ an.

2. Markieren Sie im linken Bereich die DB-Datenquelle, die Sie mit einer Datenbank verknüpfen wollen.
3. Öffnen Sie im rechten Bereich das Kontextmenü und wählen Sie „Tabelleneigenschaften“. Der *Dialog „Tabelleneigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 3.6.1.3, S. 54)* öffnet sich.
4. Wählen Sie die gewünschte Tabelle aus.

   Wenn Sie noch keine Tabelle mit den entsprechenden Spalten angelegt haben, können Sie den DB-Manager zum Anlegen der Spalten benutzen.

   → Siehe *Tabellen und Spalten mit dem DB-Manager definieren (Workbench/Process Engine: Modul-Guide, Kap. 3.4, S. 50).*
5. Aktivieren Sie die Checkboxen für die beiden Spalten „Eingabe“ und „Ausgabe“.
6. Klicken Sie auf „OK“, um die Eingabe zu speichern.
7. Öffnen Sie im rechten Bereich das Kontextmenü „Hinzufügen“, um Daten aus der Datenbank in die Tabelle zu lesen.

   Die gelesenen Daten dienen nur zur Überprüfung der Verbindung zur korrekten Datenbank. Bei größeren Datenmengen werden daher nicht alle Daten gelesen und angezeigt.
8. Prüfen Sie, ob die Verbindung zu Datenbanktabelle korrekt ist und die Daten korrekt gelesen wurden.

3.4 Tabellen und Spalten mit dem DB-Manager definieren

Voraussetzungen

- Sie haben eine Lookup Table zum Bearbeiten geöffnet.
- Sie haben eine DB-Datenquelle angelegt.

Neue Tabelle anlegen

So gehen Sie vor

1. Zeigen Sie das Register „Datenquellen“ an.
2. Wählen Sie die gewünschte DB-Datenquelle aus.
3. Öffnen Sie im rechten Bereich das Kontextmenü und wählen Sie „Tabelleneigenschaften“.
4. Klicken Sie auf den „DB-Manager“-Button. Der *Dialog „Datenbankmanager“ (Process Engine: Administrator- und Entwickler-Guide, Kap. 8.4, S. 117)* öffnet sich.

5. Legen Sie eine neue Tabelle mit den beiden Spalten „Eingabe“ und „Ausgabe“ mit den für Ihre Eingangsnachricht und Ausgangsnachricht erforderlichen Attributen (Typ, Länge, Null, Standardwert).
6. Klicken Sie auf „OK“, um die Spaltendefinitionen zu speichern.

Bei Verwendung des Moduls in einem Technical Workflow können Sie den Tabellennamen und die Spaltennamen sowie die DB-Verbindungsparameter mit Hilfe des Variablen-Mappings überschreiben.

3.5 Ersetzungsvorschriften definieren

Voraussetzungen

- Sie haben eine Lookup Table zum Bearbeiten geöffnet.
- Sie haben eine Datenquelle angelegt.
- Sie haben der XML-Datenquelle bzw. der referenzierten Datenbank Werte für die Spalten „Eingabe“ und „Ausgabe“ hinzugefügt.

So gehen Sie vor

1. Zeigen Sie das Register „Verarbeitungsvorschriften“ an.
2. Öffnen Sie im Bereich „Quelldokument“ eine beispielhafte Eingangsnachricht.
3. Öffnen Sie im Bereich „Ersetzungsvorschriften“ das Kontextmenü und wählen Sie „Hinzufügen“. Eine neue Zeile wird angelegt.
4. Doppelklicken Sie in der neuen Zeile in der Spalte „XPath“ in das Feld, um einen XPath-Abfrage zu definieren, mit der Sie den betreffenden Wert in der Eingangsnachricht suchen.
5. Wählen Sie in die passende Datenquelle.
6. Aktivieren Sie die Checkboxen, um das Verhalten der Ersetzungsvorschriften zu beeinflussen.
   → Siehe *Register „Verarbeitungsvorschriften“ (Workbench/ Process Engine: Modul-Guide, Kap. 3.6.2, S. 55).*

3.6 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Register „Datenquellen“, S. 52*
- *Register „Verarbeitungsvorschriften“, S. 55*
- *Dialog „Eigenschaften der Verarbeitungsvorschrift“, S. 56*
- *Dialog „Tabelleneigenschaften“, S. 54*

3.6.1 Register „Datenquellen“

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Datenquelle hinzufügen“ (1), S. 53*
- *Dialog „Datenquelle hinzufügen“ (2), S. 53*
- *Dialog „Tabelleneigenschaften“, S. 54*

Im Register „Datenquellen“ verwalten Sie statische XML-Datenquellen und dynamische DB-Datenquellen. Das Register besteht aus folgenden Bereichen:

- **Linker Bereich: Liste aller Datenquellen**

  Zeigt eine Liste aller bereits vorhandenen Datenquellen an.

  → Siehe *Datenquelle anlegen (Workbench/Process Engine: Modul-Guide, Kap. 3.2, S. 45).*

- **Rechter Bereich: Datenansicht**

  Zeigt die Spaltenstruktur der im linken Bereich ausgewählten Datenquelle an.

  Über das Kontextmenü können Sie die Spaltenstruktur einer XML-Datenquelle ändern oder die Tabelleneigenschaften einer DB-Datenquelle anpassen.

  Das Hinzufügen, Ändern und Verschieben von Werten sowie das Importieren von Tabellendaten ist hier nur für XML-Datenquellen möglich.

  Die Werte einer DB-Datenquelle ändern Sie direkt in der Datenbanktabelle.

  → Siehe

  - *Daten in eine XML-Datenquelle importieren (Workbench/ Process Engine: Modul-Guide, Kap. 3.2.2, S. 46)*
  - *Daten manuell zu einer XML-Datenquelle hinzufügen (Workbench/Process Engine: Modul-Guide, Kap. 3.2.3, S. 47)*
  - *Datenbanktabelle referenzieren, Tabellenname und Spaltennamen setzen (Workbench/Process Engine: Modul-Guide, Kap. 3.3, S. 49)*

3.6.1.1 Dialog „Datenquelle hinzufügen“ (1)

In diesem Dialog wählen Sie den Typ der neuen Datenquelle aus und benennen die Datenquelle.

Typ

- **XML-Datenquelle**
  XML-Datenquellen erstellen Sie direkt im Complex Lookup-Modul. Alternativ können Sie auch vorhandene XML-Datenquellen importieren.
- **DB-Datenquelle**
  Eine DB-Datenquelle referenziert eine externe Datenbanktabelle.

Informationen der -Datenquelle

- **Name**
  Name der Datenquelle. Der Name kann ein anderer sein als der Name, der in der Datenbank verwendet wird.
- **Beschreibung**: Optional.

3.6.1.2 Dialog „Datenquelle hinzufügen“ (2)

(Nur bei Typ der Datenquelle = „DB-Datenquelle“)

In diesem Dialog geben Sie die Verbindungsdaten zur Datenbank ein.

Datenquelle

- **Datenbank**
  Wählen Sie einen Datenbank- oder Business Object Connector aus, dessen aktuelle Verbindungsparameter kopiert und zum Herstellen der Datenbankverbindung genutzt werden sollen. Sie können diese Werte nachträglich ändern.

Datenbankverbindung

- **Voreinstellung**
  Zur Auswahl eines Datenbanktyps. Auf Basis der Auswahl werden in den folgenden Feldern eine Datenbank-Muster-URL und der korrekte Datenbank-Treiber gesetzt.
  Wählen Sie die Option „Custom“, um keine Vorgaben zu setzen.
- **Datenbank-URL**
  Wenn im Feld „Voreinstellung“ eine Auswahl getroffen wurde, dann wird eine korrekte Muster-URL zum Ansprechen der Datenbank angezeigt.
  Um eine andere Datenbank anzusprechen, als unter „Voreinstellung“ angegeben, ignorieren Sie die Voreinstellung und geben die korrekte URL Ihrer Datenbank ein.

- **Datenbanktreiber-Klasse**
  Name der Datenbanktreiber-Klasse. Wird automatisch korrekt gesetzt, wenn unter „Voreinstellung“ eine Datenbank ausgewählt wurde.

Der zur gewählten Datenbank passende Datenbanktreiber muss installiert sein! Siehe *Treiber installieren (Process Engine: Administrator- und Entwickler-Guide, Kap. 3.4, S. 51).*

- **Benutzer**
  Gültiger Benutzername für die ausgewählte Datenbank.
- **Passwort**
  Passwort, mit dem sich der Benutzer gegenüber der Datenbank authentifiziert.
- **Textwerte aus der Datenbank trimmen**
  Löscht bei Textwerten führende und nachfolgende Leerstellen.
- **Spezielle Kodierung setzen**
  Wenn markiert, dann können Sie einen Zeichensatz auswählen oder einen beliebigen Zeichensatz eingeben.
- **Connection Pooling**
  Öffnet den *Dialog „Datenbank Connection Pooling“ (Workbench/ Process Engine: Systemkonnektor-Guide, Kap. 6.5.5, S. 87)*, in dem Sie die Wiederverwendung physischer Datenbankverbindungen konfigurieren, um den zeitaufwändigen Auf- und Abbau von Datenbankverbindung zu optimieren.
- **Spezielle Verbindungsparameter**
  Erstellen Sie hier Parameter (z. B. Timeouts, Anzahl offener Verbindungen etc.), mit denen Sie die Datenbankverbindung abhängig von der jeweils verwendeten Datenbank konfigurieren. Die Parameter definieren Sie als Name-Wert-Paare.

Verbindungstest

- **Verbindung testen**
  Zum Testen, ob die Verbindung mit Ihren Angaben erfolgreich aufgebaut werden kann.

3.6.1.3 Dialog „Tabelleneigenschaften“

In diesem Dialog definieren Sie die Eigenschaften der Tabellen einer DB-Datenquelle.

- **Tabelle**
  Zur Auswahl einer Tabelle aus der Liste. Sie müssen die Tabelle zuvor in der Datenbank angelegt haben.

Die Namen der DB-Tabelle, der Eingabe- und Ausgabespalte können dynamisch über das Variablen-Mapping überschrieben werden. Wählen Sie dazu als Ziel „Moduleigenschaft“ aus und geben Sie dann den Namen der Moduleigenschaft ein, die überschrieben werden soll:

```
- DB: <Tabellenname>.SQL_Table
- DB: <Tabellenname>.SQL_ColIn
- DB: <Tabellenname>.SQL_ColOut
```

- **Button „DB-Manager“**
  Zum Anlegen einer neuen Tabelle und der erforderlichen Spalten.
  → Siehe *Dialog „Datenbankmanager“ (Process Engine: Administrator- und Entwickler-Guide, Kap. 8.4, S. 117).*

Spaltenstruktur

Aktivieren Sie in der Zeile mit dem Spaltennamen für die Eingabewerte die Checkbox in der Spalte „Eingabe“ und in der Zeile mit dem Spaltennamen für die Ausgabewerte die Checkbox in der Spalte „Ausgabe“.

3.6.2 Register „Verarbeitungsvorschriften“

Im Register „Verarbeitungsvorschriften“ erstellen und konfigurieren Sie Verarbeitungsvorschriften. Das Register besteht aus folgenden Bereichen:

- **Vorschrift**
  Zum Bearbeiten, Testen und Überprüfen der Vorschrift:
  - **Test ausführen**:
    Zum Testen der Verarbeitungsvorschriften. Das Ergebnis wird im Bereich „Ergebnis“ angezeigt. Wurden alle Ersetzungen korrekt ausgeführt, können Sie die Modul-Konfiguration speichern und das Modul publizieren.
  - **Vorschrift überprüfen**
    Zum Überprüfen der Gültigkeit der Verarbeitungsvorschriften.
- **Ersetzungsvorschriften**
  Zum Anlegen der einzelnen Ersetzungsvorschriften.
  In der Tabelle ordnen Sie ausgewählten XML-Elementen der Eingangsnachricht jeweils eine Datenquelle zu. Für jede Zuordnung können Sie folgende Infos angeben:
  - **XPath**
    Zur Auswahl eines XML-Knotens doppelklicken Sie in der Spalte „XPath“ in eine Tabellenzeile. Ein Klick auf den -Button öffnet den XPath-Assistenten.

- **Datenquelle**
  Zum Auswählen einer Datenquelle klicken Sie in das Feld und wählen eine Tabelle aus.
  Sie können jeder Ersetzungsvorschrift eine eigene Datenquelle zuordnen.
- **Aktiv**: Zeigt an, ob diese Zeile mit verarbeitet werden soll.
- **Groß-/Kleinschreibung**
  Wenn markiert, dann wird bei der Ersetzung der Werte die Groß- und Kleinschreibung beachtet.
- **Invertieren**
  Legt fest, dass die Datenquelle in umgekehrter Reihenfolge benutzt werden soll, d. h. in den eingehenden Nachrichten wird nach den Werten gesucht, die in der Datenquelle in der Spalte „Ausgabe“ stehen, diese werden dann auf die Werte der Spalte „Eingabe“ abgebildet.

- **Quelldokument**
  Zum Öffnen und Anzeigen einer Beispiel-Eingangsnachricht.
  Um die XPath-Angaben im Bereich „Aktionslisten“ zu erstellen, wird die Eingangsnachricht automatisch im XPath-Assistenten geöffnet.
- **Namensräume**
  In diesem Bereich werden die Namensräume angezeigt, die in der Eingangsnachricht definiert sind. Sie können Namensraumdefinitionen hinzufügen, ändern und löschen.
- **Ergebnis**
  In diesem Bereich wird das Ergebnis angezeigt, wenn Sie die Verarbeitungsvorschriften getestet haben.

3.6.2.1 Dialog „Eigenschaften der Verarbeitungsvorschrift“

Aufruf

Register „Verarbeitungsvorschriften“ > Doppelklick auf den Eintrag „Vorschrift“

Verarbeitungsvorschrift

- **Name**
  Der angezeigte Name ist fix und kann nicht geändert werden.

Eigenschaften der Verarbeitungsvorschrift

- **Fehler anzeigen, wenn kein Wert gefunden wurde**

Wenn diese Option aktiviert ist und bei der Ausführung des Moduls in der Tabelle kein passender Wert zu einem XML-Element in der dazugehörigen Datenquelle gefunden wurde, dann wird die Ausführung mit einer Fehlermeldung abgebrochen.

Verwendete Datenquellen

Die Tabelle listet die definierten Datenquellen mit Name, Typ, Kommentar und Status (aktiv/inaktiv) auf. man kann auch welche anlegen

4 Task Generator (Data Converter)

Dieser Abschnitt erläutert die folgenden Themen:

- *Funktionsprinzip des Task Generators, S. 60*
- *Formulare im Formular Designer erstellen, S. 63*
- *Benutzereingaben in Formularen validieren, S. 80*
- *Kontextsensitive Onlinehilfe erstellen, S. 84*
- *Formular-Texte zum Übersetzen exportieren und importieren, S. 87*
- *Widgets verwenden und erstellen, S. 90*
- *Mitarbeiter mit Tasks in Workflows einbeziehen, S. 100*
- *Benutzeroberfläche des Formular Designers, S. 107*
- *Benutzeroberfläche des Client Generators, S. 120*
- *Dialogbeschreibungen, S. 124*

Verwendung

Der Task Generator ist ein wichtiges Werkzeug, um Mitarbeiter in automatisierte Prozesse einzubeziehen. Mit einem Task Generator können Sie Tasks und Web Applikationen erstellen:

- **Tasks**

  Tasks sind Aufgaben, die während der Abarbeitung von Geschäftsprozessen anfallen und einen manuellen Eingriff erfordern. Eine typische Task ist z. B. die Freigabe einer Bestellanforderung.

  Wenn Sie einen Mitarbeiter über die Zuweisung einer Task in einen Prozess einbinden, wird ein Eintrag in der individuellen Aufgabenliste des Mitarbeiters erstellt. Die Aufgabenliste kann in einem Web-Browser, im inubit Enterprise Portal oder in der inubit Workbench angezeigt werden.

  Tasks können rollen- oder regelbasiert zugewiesen werden.

  Der Zeitrahmen, in dem die Tasks abgearbeitet werden müssen, wird beim Erzeugen der Tasks festgelegt. Falls eine Task nicht wie geplant erledigt wird, kann ein Eskalations- oder Delegationsmechanismus angestoßen werden.

  Die verschiedenen Tasktypen der inubit Suite 6 ermöglichen es, Informationen anzuzeigen, Aufgaben mit Hilfe externer Applikationen zu erledigen, Reports oder beliebige, andere Dokumente in die Aufgaben einzubinden und Daten in Formulare einzugeben.

  → Siehe *Mitarbeiter mit Tasks in Workflows einbeziehen (Workbench/Process Engine: Modul-Guide, Kap. 4.7, S. 100).*

- **Web Applikationen**

Wenn der Prozess durch eine Benutzerinteraktion gestartet werden soll, können Sie mit einem Task Generator ein Formular für eine Web Applikation erstellen. Web Applikationen sind komplette, formularbasierte Anwendungen im inubit Enterprise Portal.

→ Siehe
- *Eigene Web-Applikation entwickeln (Tutorials, Kap. 7.2, S. 132)*
- *inubit Enterprise Portal verwenden (Workbench: Benutzer-Guide, Kap. 20, S. 505)*

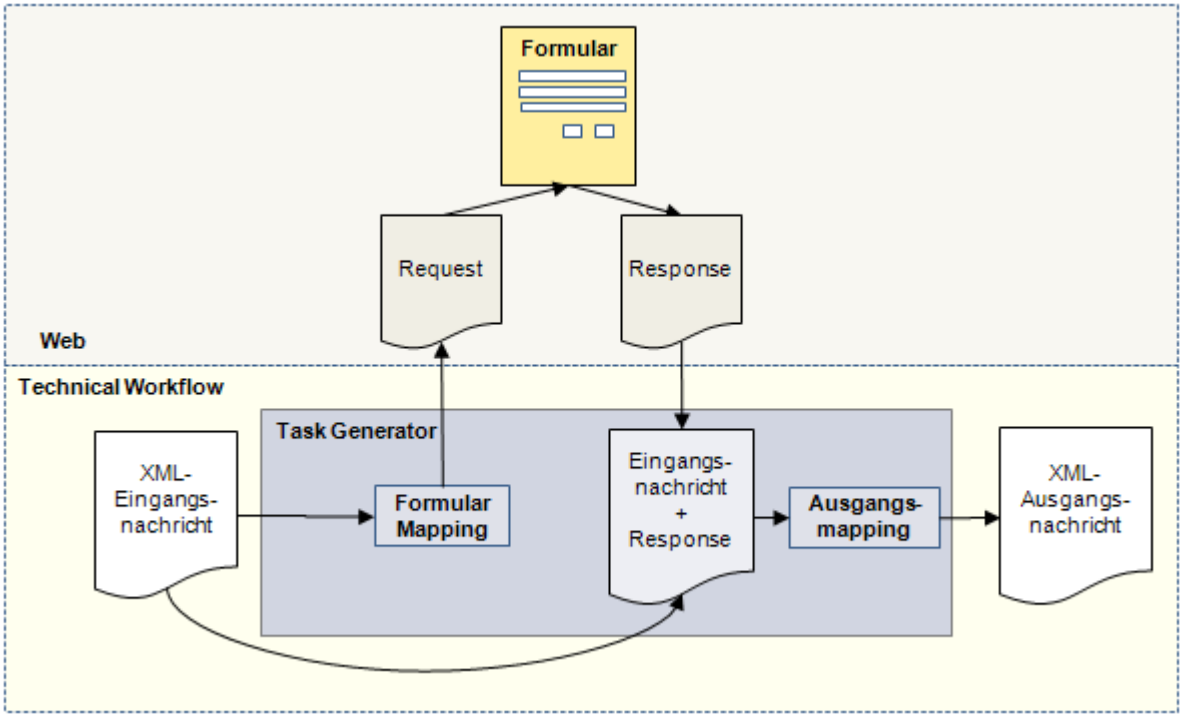
Features des Task Generators

Die folgenden Features des Task Generators sind in allen Formularen nutzbar, unabhängig davon, ob es sich dabei um Formulare in Web Applikationen handelt oder um Tasks vom Typ „Formular“:

- Gültigkeit von Benutzereingaben client- oder server-seitig prüfen lassen.
  → Siehe
  - *Clientseitige Validierung von Benutzereingaben über Pattern (Workbench/Process Engine: Modul-Guide, Kap. 4.3.1, S. 81)*
  - *Server-seitige Validierung von Benutzereingaben per AJAX (Workbench/Process Engine: Modul-Guide, Kap. 4.3.2, S. 82)*
- Kontextsensitive Hilfe zu Formularfeldern hinterlegen.
  → Siehe *Kontextsensitive Onlinehilfe erstellen (Workbench/Process Engine: Modul-Guide, Kap. 4.4, S. 84).*
- Mehrsprachige Hilfe- und Formulartexte nutzen.
  → Siehe *Formular-Texte zum Übersetzen exportieren und importieren (Workbench/Process Engine: Modul-Guide, Kap. 4.5, S. 87).*
- Widgets nutzen.
  → Siehe *Widgets verwenden und erstellen (Workbench/Process Engine: Modul-Guide, Kap. 4.6, S. 90).*
- Formulare dynamisch generieren.
  → Siehe *Formularteile aus XML Schemas/WSDL-Dateien mit dem Client Generator erzeugen (Workbench/Process Engine: Modul-Guide, Kap. 4.2.8, S. 72).*

4.1 Funktionsprinzip des Task Generators

Ein Task Generator, der ein Formular-Task oder ein Formular für eine Web Applikation erzeugt, funktioniert nach folgendem Prinzip:



In dem Task Generator werden zwei Konvertierungsschritte nacheinander durchgeführt:

1. Die Daten aus einer Eingangsnachricht werden auf ein Formular abgebildet. Danach wird das Formular von Benutzern bearbeitet.
2. Das Ergebnis der Formularbearbeitung geht zurück an den Task Generator und wird so aufbereitet, wie es das nachfolgende Modul erwartet.

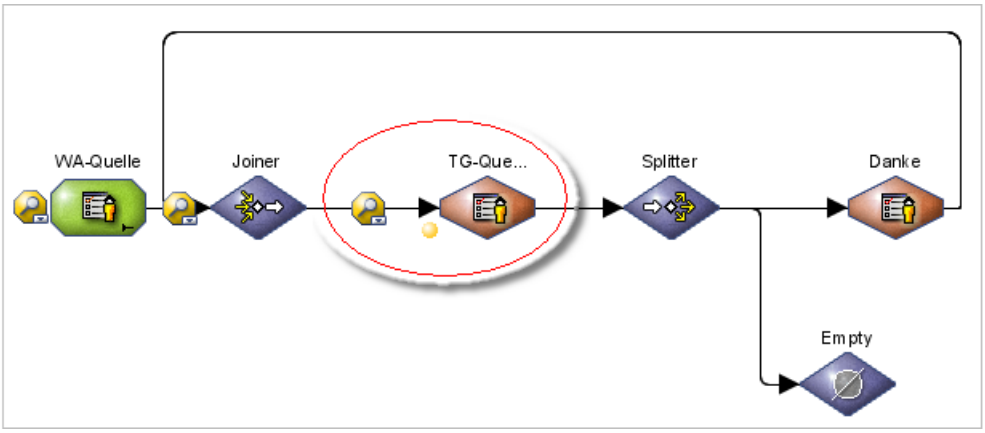
Im Detail sehen die Konvertierungsschritte so aus:

1. Im Formular Mapping wird aus der Eingangsnachricht mit einem XSLT-Stylesheet ein Request erzeugt.
2. Aus dem Request wird ein Formular zur Anzeige in der inubit Workbench oder ein HTML-Request für einen Browser erzeugt. Aus dem HTML-Request erzeugt der Browser ein Formular.
3. Das Formular wird von den Benutzern bearbeitet. Mit einem Klick auf den Submit-Button werden die Formulardaten als Response an den Task Generator zurückgesendet.
4. Die Response wird mit der Eingangsnachricht zusammengeführt und im Ausgangsmapping auf das XML-Format abgebildet, welches das nächste Modul im Workflow erwartet.

Watchpoints im Designer

Die Eingangsnachrichten der beiden Konvertierungsschritte sind auch im Watch-Modus des Designers sichtbar:

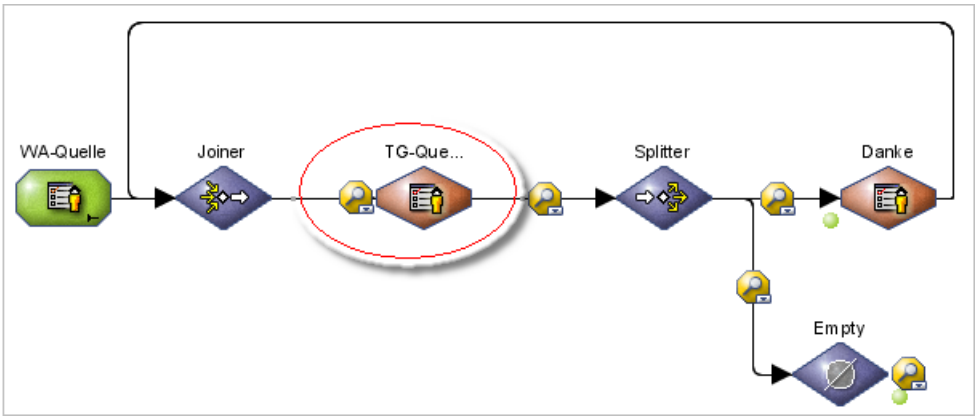
1. Solange das Formular angezeigt wird, steht ein Watchpoint deutlich vor dem Verbindungspfeil (hier: vor dem Modul „TG-Quelle“):



Wenn Sie den Watchpoint öffnen, wird die Eingangsnachricht des Task Generators angezeigt.

Der Workflow wartet nun solange, bis der Browser die Formulardaten zurücksendet, d. h., bis ein Benutzer den Submit-Button des Formulars drückt.

2. Nachdem der Submit-Button gedrückt wurde, wird die Response an den Task Generator gesendet und das nächste Formular angezeigt. Der Watchpoint wird nun kurz vor dem Task Generator auf der Pfeilspitze (hier: vor dem Modul „TG-Quelle“) angezeigt:



Wenn Sie den Watchpoint öffnen, dann wird die Response angezeigt.

Die Response wird im Task Generator im Ausgangsmapping nicht in dieser Form angezeigt. Dort wird sie direkt mit der Eingangsnachricht zusammengeführt.
Die Response benötigen Sie z. B., um bei der Inter-Portlet-Kommunikation die übergebenen Daten in dem Formular abzubilden. Siehe *Daten zwischen Web-Applikationen austauschen: Inter-Portlet-Kommunikation (Workbench: Benutzer-Guide, Kap. 20.17, S. 544).*

4.2 Formulare im Formular Designer erstellen

Dieser Abschnitt erläutert die folgenden Themen:

- *Formular-Layout festlegen und Elemente auf Panels positionieren, S. 63*
- *Formularvorschau anzeigen, S. 65*
- *Formulare mit CSS gestalten, S. 65*
- *Combo-Boxen erstellen, S. 68*
- *Liste mit Mehrfachauswahl dynamisch erstellen, S. 70*
- *Maßeinheit für Größenangaben ändern, S. 71*
- *Formularelemente abhängig von Rechten machen (nur bei Web-Applikationen), S. 71*
- *Formularteile aus XML Schemas/WSDL-Dateien mit dem Client Generator erzeugen, S. 72*
- *JavaScript in Formularen verwenden, S. 74*
- *Portlets und Formularabschnitte mit Ajax aktualisieren, S. 76*

Eine praktische Einführung in das Arbeiten mit dem Formular-Designer finden Sie im Tutorial *Web-Applikation in inubit Workbench entwickeln (Tutorials, Kap. 7, S. 131)* in den Abschnitten *Formular-Mapping erstellen (Tutorials, Kap. 7.2.1, S. 137)* und *Ausgangs-Mapping erstellen (Tutorials, Kap. 7.2.2, S. 141)*.

4.2.1 Formular-Layout festlegen und Elemente auf Panels positionieren

Bereits beim Konfigurieren eines Task Generators legen Sie mit dem so genannten Panel-Layout das Basis-Layout Ihres Formulars fest.

Nach dem Fertigstellen des Task Generators wird der Formular Designer angezeigt, in dem Sie per Drag'n'Drop die verschiedenen Formular-Elemente wie z. B. Eingabefelder, Label und Buttons zu einem Formular zusammenstellen.

→ Siehe

- *Dialog „Panel Layout" (Workbench/Process Engine: Modul-Guide, Kap. 4.10.5, S. 130)*
- *Benutzeroberfläche des Formular Designers (Workbench/ Process Engine: Modul-Guide, Kap. 4.8, S. 107)*
- *Formular-Elemente in der Werkzeugleiste (Workbench/Process Engine: Modul-Guide, Kap. 4.8.3, S. 111)*
- *Eigenschaften von Formular-Elementen (Workbench/Process Engine: Modul-Guide, Kap. 4.8.4, S. 114)*

So gehen Sie vor

1. **Mehrere Panels verwenden**

   Initial wird im Formular Designer eine leere Fläche angezeigt. Diese Fläche ist das erste Panel des Formulars namens „main“. Ein Formular kann aus einem oder mehreren Panels bestehen.

   a. Panel anlegen:

   Klicken Sie auf den Button „Panel“:

   Ein Dialog zum Benennen des Panels öffnet sich. Benennen Sie das Panel und schließen Sie den Dialog mit „OK“.

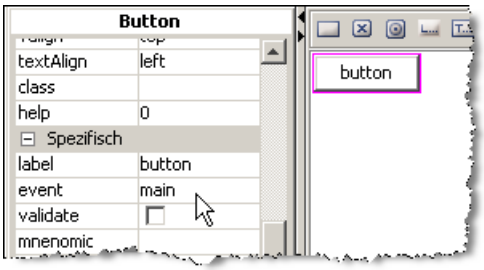
   Das neue, leere Panel wird angezeigt.

   b. Um im Formular Designer zwischen Panels zu wechseln, nutzen Sie die Panel-Liste:

   c. Im Formular zwischen Panels wechseln:

   Fügen Sie einen Button in das Panel ein. Ein Klick auf den Button soll ein ausgewähltes Panel anzeigen.

   Geben Sie den Namen des Panels, das angezeigt werden soll, im Register „Eigenschaften“ bei dem Button-Attribut `event` ein, z. B.:

2. **Formular-Elemente positionieren**

   a. Ziehen Sie ein Formular-Element aus der Werkzeugleiste des Formular-Designers auf das darunter liegende Panel. Es wird in seiner Standard-Konfiguration angezeigt.

   b. Ziehen Sie ein weiteres Formular Element auf das Panel. Ziehen Sie das Element mit der Maus genau an die Seite des bereits vorhandenen Elements, an der es positioniert werden soll.

   Eine rote Linie zeigt an, wo das Element eingefügt wird:

	Rechts neben dem Formular-Element

	Links neben dem Formular-Element
	Über dem Formular-Element
	Unter dem Formular-Element

4.2.2 Formularvorschau anzeigen

Sie können in jeder Entwicklungsphase eine Vorschau auf Ihr Formular anzeigen lassen:

So gehen Sie vor

- **Vorschau in der inubit Workbench**
  Für Formulare, die in der inubit Workbench im Register „Taskliste“ angezeigt werden sollen.
  Klicken Sie in der Werkzeugleiste des Formular Designers auf den -Button.
- **Vorschau im Browser**
  Für Formulare, die in einem Browser angezeigt werden sollen.
  Klicken Sie in der Werkzeugleiste des Formular Designers auf den -Button.
  Wenn im *Dialog „Berechtigungen“ (Workbench/Process Engine: Modul-Guide, Kap. 4.10.2, S. 127)* des Task Generators ein Web Application Connector ausgewählt ist, dann werden in der Vorschau die CSS-Stylesheets und JavaScript-Skripte des inubit Enterprise Portals und des Web Application Connectors berücksichtigt.

Es hängt von den Einstellungen Ihres Applikationsprofils ab, in welchem Browser Ihr Formular angezeigt wird. Siehe *Applikationsprofile (Process Engine: Administrator- und Entwickler-Guide, Kap. 11.1, S. 135)*.

4.2.3 Formulare mit CSS gestalten

Dieser Abschnitt erläutert die folgenden Themen:

- *CSS zentral definieren, S. 66*
- *CSS lokal definieren, S. 66*
- *Alternativen, S. 67*

4.2.3.1 CSS zentral definieren

Sie können für Formularelemente CSS-Klassen anlegen und die Eigenschaften dieser Klassen in einem Web Application Connector definieren.

Dieses Vorgehen hat den Vorteil, dass Sie die CSS-Klassen in allen Task Generator-Formularen nutzen können, die den Web Application Connector referenzieren.

So gehen Sie vor

1. Markieren Sie das Element im Formular Designer.
2. Um das Element mit einer CSS-Klasse zu verknüpfen, zeigen Sie das Register „Eigenschaften“ an und geben beim Attribut `class` den Namen der Klasse ein.
3. Legen Sie einen Web Application Connector an und definieren Sie im *Dialog „Interne Ressourcen“ (Workbench/Process Engine: Systemkonnektor-Guide, Kap. 39.3.2, S. 403)* die Eigenschaften der Klasse.

Informationen über die Syntax von Formatdefinitionen für Klassen finden Sie unter http://de.selfhtml.org/css/formate/zentrale.htm#klassen.

4. Publizieren Sie den Web Application Connector.

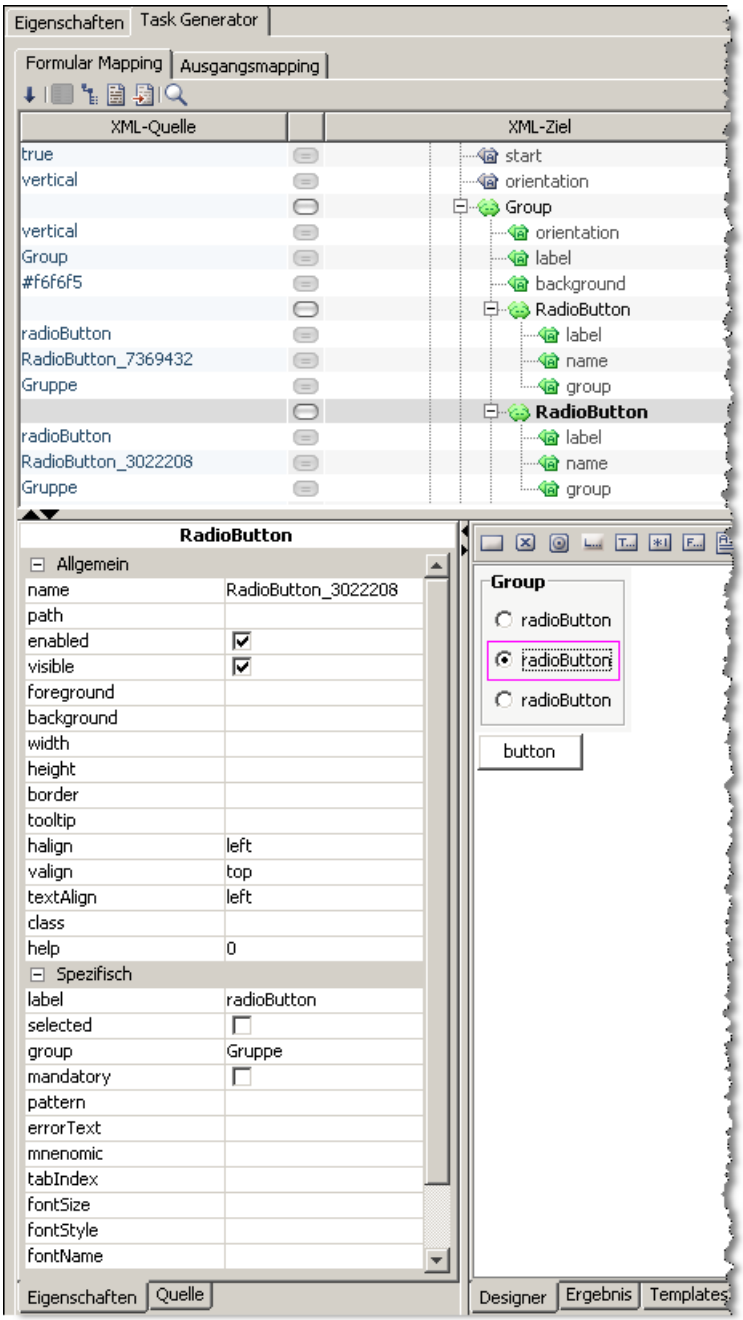
Der Web Application Connector muss publiziert und deployed werden, aber nicht Teil eines Workflows sein!

5. Zeigen Sie den *Dialog „Berechtigungen“ (Workbench/Process Engine: Modul-Guide, Kap. 4.10.2, S. 127)* des Task Generators an und referenzieren Sie den Web Application Connector.

4.2.3.2 CSS lokal definieren

So gehen Sie vor

1. Markieren Sie das Formularelement im Designer oder im XML Ziel:



2. Füllen Sie im Register „Eigenschaften“ des Formular Designers die Attribute des Formularelements mit Werten.

4.2.3.3 Alternativen

Falls Sie keine CSS-Klassen im Web Application Connector einsetzen möchten und die gewünschten Attribute im Register „Eigenschaften“ des Formular Designers nicht vorhanden sind, gibt es zusätzlich folgende Optionen:

- **Attribut `style` manuell zum Formularelement hinzufügen**

Beispiel: `<Button label="next" event="submit(OK) style="text-align: center; color: black; font-family: arial"/>`

Dieses Vorgehen ist z. B. geeignet, um Attribute aus dem Register “Eigenschaften“ zu ergänzen.

- **Attribut `css` zum Formularelement hinzufügen**

  Beispiel: `<Button label="next" event="submit(OK) css="text-align: center; color: black; font-family: arial"/>`

  Bei diesem Vorgehen werden nur die Stile verwendet, die über das Attribut `css` definiert sind. Die Attributwerte aus dem Register “Eigenschaften“ werden ignoriert.

4.2.4 Combo-Boxen erstellen

Bei Combo-Boxen können Formularbenutzer einen Wert aus einer festen Menge von Werten auswählen oder einen beliebigen Wert eingeben, z. B.:

Das Eingeben eines beliebigen Wertes funktioniert nur in Formularen, die in der Taskliste der inubit Workbench angezeigt werden!

So gehen Sie vor

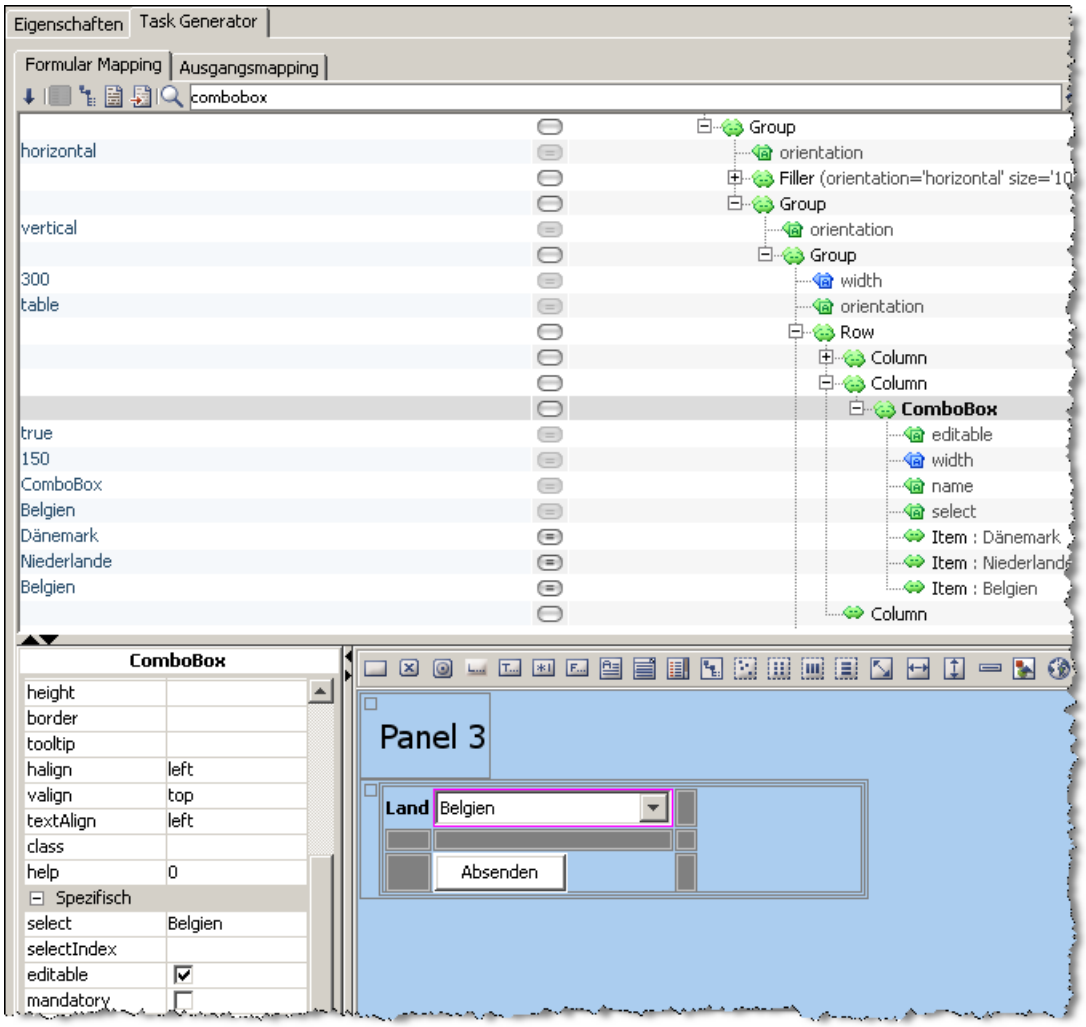
1. **Combo-Box erstellen**
   a. Fügen Sie im Formular Designer ein `ComboBox`-Element ein.
   b. Fügen Sie für jeden Wert, der in der Liste angezeigt werden soll, ein `Item`-Element ein.
   c. Um einen internen Wert zu übergeben (zusätzlich zu dem Wert, den die Benutzer auswählen), setzen Sie bei jedem `Item`-Element das Attribut `value`.
2. **Eingabe in Combo-Box ermöglichen**

Nur bei Formularen, die in der Taskliste der inubit Workbench angezeigt werden.

Markieren Sie das `ComboBox`-Element im Designer und setzen Sie bei der Eigenschaft `editable` ein Häkchen. Wenn das Häkchen nicht gesetzt ist, dann können Benutzer nur einen der vorgegebenen Werte aus der Liste wählen.

3. **Standardwert setzen**

   Um einen Standardwert vorzugeben, der beim Anzeigen der Liste bereits ausgewählt ist, markieren Sie das `ComboBox`-Element und geben den Standardwert als Wert der Eigenschaft `select` ein, z. B.:



Alternativ können Sie einen Standardwert auch mit folgenden Attributen setzen:

- `selectIndex='[Ziffer]'` bei dem `ComboBox`-Element, wenn die Reihenfolge der `Item`-Einträge bekannt ist.
- `selected='true'` bei einem `Item`-Element, wenn die `Item`-Einträge dynamisch generiert werden

4.2.5 Liste mit Mehrfachauswahl dynamisch erstellen

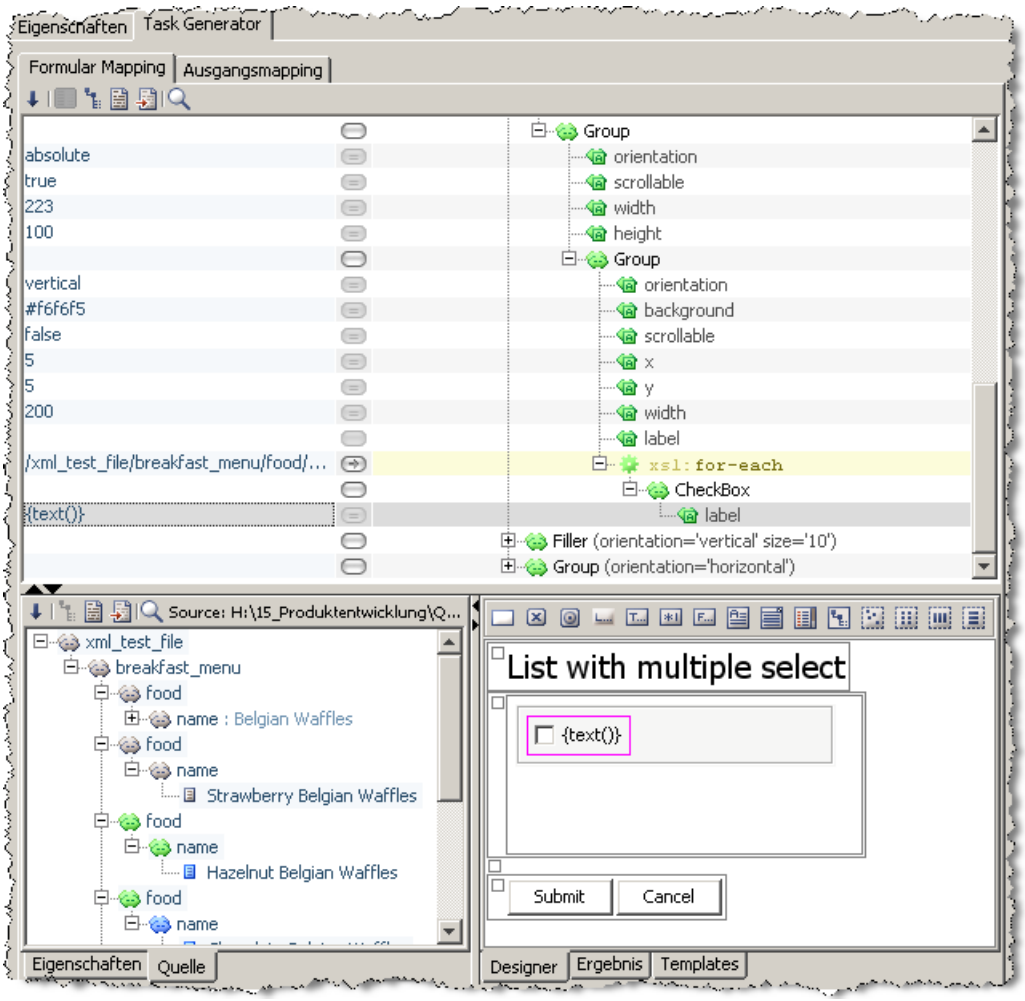
Aus einer Liste mit Mehrfachauswahl können Formularbenutzer einen oder mehrere Werte aus einer festen Menge von Werten auswählen.

Voraussetzungen

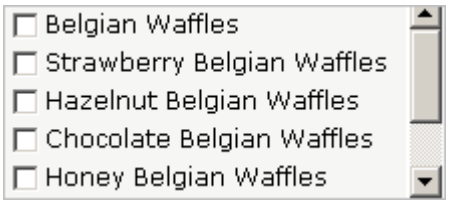
Eingangsnachricht mit einer Werteliste, um die Formular-Liste dynamisch generieren zu lassen.

So gehen Sie vor

1. Fügen Sie im Formular Designer `group`-Element mit absolutem Layout ein.
2. Fügen Sie in diese eine Gruppe mit vertikalem Layout ein.
3. Fügen Sie in diese Gruppe wiederum mit Hilfe `xsl:for-each` so viele `CheckBox`-Elemente ein, wie Sie Listeneinträge benötigen, z. B.:



Das Ergebnis ist eine scrollbare Liste, in der jeder Eintrag ausgewählt werden kann:



4.2.6 Maßeinheit für Größenangaben ändern

Standardmäßig wird die Einheit „pt“ (Punkt) für alle Größenangaben in Formularen verwendet. Sie können diese Einheit ändern.

So gehen Sie vor

1. Fügen Sie dem Element `Form` das Attribut `metric` hinzu.
2. Geben Sie als Wert die gewünschte Einheit ein, z. B. „px“.

4.2.7 Formularelemente abhängig von Rechten machen (nur bei Web-Applikationen)

In Formularen für Web Applikationen (Tasktyp „Direkte Formularfolge (Seite einer Web-Applikation)“) können Sie für Formularelemente wie z. B. Buttons, Eingabefeldern oder DropDown-Listen festlegen, welche Rechte nötig sind, um die Formularelemente anzuzeigen und/oder zu bearbeiten.

Die Rechte werden im inubit Enterprise Portal zu Portalrollen zugeordnet und die Portalrollen zu Portal-Benutzern. So können Sie detailliert steuern, welche Benutzer zum Anzeigen und Bearbeiten der Formularelemente berechtigt sind.

Voraussetzungen

- Der Task Generator, welcher das Formular erzeugt, muss Teil eines Technical Workflows sein, der über einen Web Application Connector als Portlet auf das inubit Enterprise Portal deployed wird.
- Sie haben bereits ein Formular erstellt.

So gehen Sie vor

1. Definieren Sie beim Konfigurieren des Web Application Connector alle nötigen Applikationsrechte.
2. Publizieren Sie den Web Application Connector.

3. Öffnen Sie das Task Generator Modul zum Bearbeiten und legen Sie fest, aus welchem Web Application Connector die Applikationsrechte übernommen werden sollen.
4. Zeigen Sie das Formular im Formular Designer an.
5. Markieren Sie ein Formularelement, das Sie abhängig von Applikationsrechten machen möchten. Im Register „Eigenschaften“ werden alle Attribute des markierten Elements angezeigt.
6. Um festzulegen, dass das Element nur für Inhaber bestimmter Rechte sichtbar ist, doppelklicken Sie auf das Attribut `visible`. Ein Dialog mit einer Liste aller verfügbaren Rechte öffnet sich.
7. Markieren Sie alle Rechte, die zum Anzeigen des Elements berechtigen.
8. Um festzulegen, dass ein Element nur von Inhabern ausgewählter Rechte geändert werden darf, doppelklicken Sie das Attribut `enabled` und wählen dann die Rechte aus.

Alle Formularelemente, denen Rechte zugewiesen wurden, werden im Formular grün umrandet.

→ Siehe

- Zuordnung von Rechten zu Rollen: *Web-Applikationsrechte vergeben und zu Portal-Rollen zuordnen (Tutorials, Kap. 7.4, S. 154)*
- *Dialog „Rechteverwaltung“ (Workbench/Process Engine: Systemkonnektor-Guide, Kap. 39.3.3, S. 404)*
- *Dialog „Berechtigungen“ (Workbench/Process Engine: Modul-Guide, Kap. 4.10.2, S. 127)*

4.2.8 Formularteile aus XML Schemas/WSDL-Dateien mit dem Client Generator erzeugen

Der Task Generator bietet die Möglichkeit, Teile von Formularen schnell und komfortabel aus XML Schemas oder WSDL-Dateien zu erstellen. Dabei werden XML-Typdefinitionen wie folgt auf Formularelemente abgebildet:

XML-Element/Attribut	Formular-Elemente
complexType	Erzeugt ein OuterGroup-Element, ein Label-Element und für jedes enthaltene Element ein InnerGroup-Element
complexType mit einer sequence	Group-Element, mit einer Reihe von Feldern, deren Reihenfolge durch ihr Vorkommen im sequence-Element festgelegt ist

XML-Element/Attribut	Formular-Elemente
simpleType	Erzeugt ein group-Element, das ein Label- und ein (zum Element passendes) Formular-Element enthält; der Elementname wird im erzeugten Formular als Bezeichner des Elements (als Label) verwendet
simple Type vom Typ „enumeration“	DropDown-Liste
simple Type vom Typ „boolean“	Checkbox
minOccurs/maxOccurs	Legen die Kardinalität der Elemente fest und damit auch, wie oft ein Element im Formular vorkommen darf oder muss. Eine maxOccurs=n kann beliebig eingeschränkt werden

Voraussetzungen

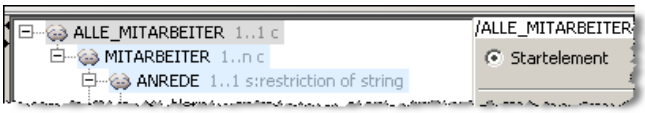
- XML Schema:
  - Das XML Schema ist gültig.
  - Das XML Schema ist im Repository gespeichert.
- WSDL-Datei:
  Die WSDL-Datei enthält einen `Types`-Abschnitt mit voll spezifizierten Definitionen.

So gehen Sie vor

1. Erstellen Sie einen Task Generator.
2. Zeigen Sie den Formular Editor an.
3. Fügen Sie eines der folgenden Elemente ein:
   - Group - Horizontal Layout
   - Group - Vertical Layout
4. Markieren Sie das Element, sodass der Rand farbig hervorgehoben wird.
5. Öffnen Sie das Kontextmenü und wählen Sie „Client Generator öffnen“. Der Client Generator öffnet sich in einem eigenen Fenster.
6. Klicken Sie im Client Generator auf „Öffnen“. Der Repository Browser öffnet sich.
7. Navigieren Sie zu Ihrer XML Schema- bzw. WSDL-Datei, um diese zu laden. Die Datei wird angezeigt.
8. Markieren Sie das Startelement Ihres Formulars mit der Maus. Es wird grau hinterlegt.

Das Startelement muss eindeutig sein und daher eine Kardinalität von „1...1“ haben!

9. Markieren Sie die Option „Startelement“.



Damit haben Sie bereits die Mindest-Angaben gemacht, die zum Erzeugen des Formulars nötig sind.

10. Zum Anzeigen des Formulars haben Sie folgende Möglichkeiten:
    - Klicken Sie auf „Testen“. Eine Vorschau auf das Formular wird in einem eigenen Java-Fenster angezeigt. In diesem Fenster sind nicht alle Funktionen des Formulars verfügbar!
    - Um alle Funktionen zu testen, zeigen Sie das Formular in Ihrem Standard-Web-Browser an:
        a. Klicken Sie auf „OK“. Die Einstellungen werden gespeichert und der Client Generator schließt sich. Im Formular Editor werden die generierten Formularelemente innerhalb der Gruppe angezeigt, in die das Schema/die WSDL-Datei geladen wurde. Die Gruppe wird zusätzlich gelb umrandet.
        b. Klicken Sie in der Menüleiste der inubit Workbench auf . Der Browser öffnet sich und zeigt die erzeugten Formularfelder an.

Führen Sie keine manuellen Änderungen an den generierten Feldern im `group`-Element durch. Alle manuellen Änderungen werden überschrieben, wenn die Schema- bzw. WSDL-Datei neugeladen wird.

Formular-Elemente wie z. B. Buttons zum Absenden oder Abbrechen des Formulars fügen Sie per Drag’n’Drop ein, siehe *Formular-Mapping erstellen (Tutorials, Kap. 7.2.1, S. 137)*

→ Siehe *Benutzeroberfläche des Client Generators (Workbench/ Process Engine: Modul-Guide, Kap. 4.9, S. 120).*

4.2.9 JavaScript in Formularen verwenden

Die Art und Weise, wie Sie JavaScript in Formularen einsetzen können, hängt von der Verwendung des Formulars und vom JavaScript-Code ab:

- *JavaScript in Formular-Tasks/ Web Applikation (Workbench/ Process Engine: Modul-Guide, Kap. 4, S. 75)*
- *JavaScript in Web Applikationen (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 76)*

- *Javascript-Eventhandler (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 76)*

→ Die Dokumentation des inubit JavaScript-Frameworks finden Sie im Verzeichnis `<iS-installdir>/documentation/jsdoc/index.html`, wenn Sie die inubit-Dokumentation installiert haben. Oder rufen Sie die Dokumentation direkt auf.

JavaScript in Formular-Tasks/ Web Applikation

In Formularen für Tasks und Web Applikationen können Sie JavaScript-Code direkt eingeben oder als externe Datei referenzieren:

- **Direkt eingeben**

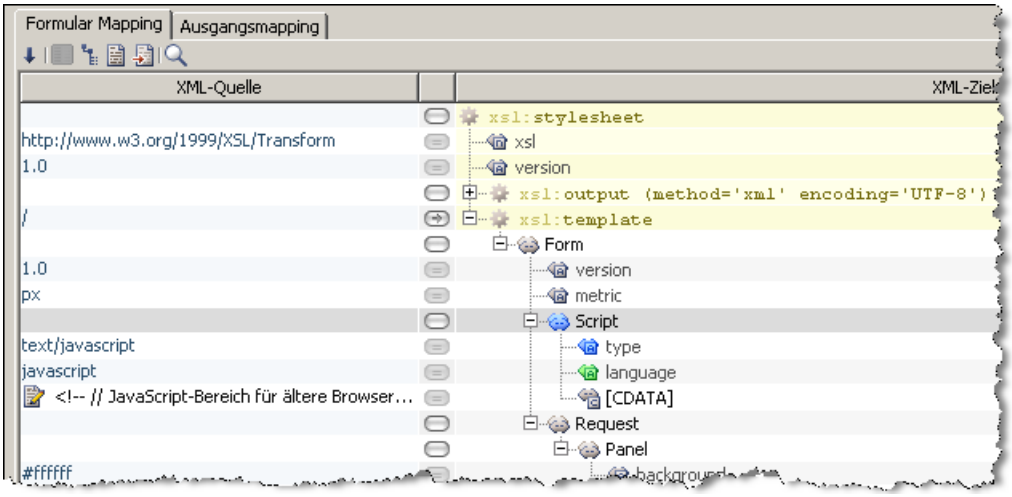
  Geben Sie den JavaScript-Code im Formular Designer in eines der folgenden Elemente ein:

  - `Web`-Element,

    → Siehe *Web (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 113)*.

  - `Script`-Element,

    Beispiel:



- **Als externe Datei referenzieren**

  a. Speichern Sie die `*.js`-Datei mit dem JavaScript-Code in einem Verzeichnis unterhalb von `<iS-installdir>/server/Tomcat/webapps/ROOT`.

  b. Fügen Sie ein `Script`-Element in das Stylesheet ein.

  c. Geben Sie im Attribut `src` den Pfad der `*.js`-Datei an, z. B. `<Script language="javascript" src="/myPortal/myJavaScript.js"/>`.

  Wenn das Formular im Browser angezeigt wird, wird die JavaScript-Datei unter `<iS-installdir>/server/Tomcat/webapps/ROOT/<myPortal>/<myJavascript.js>` gesucht.

Externer JavaScript-Code wird beim Anzeigen der Formular-Vorschau im Browser nicht ausgeführt!

JavaScript in Web Applikationen

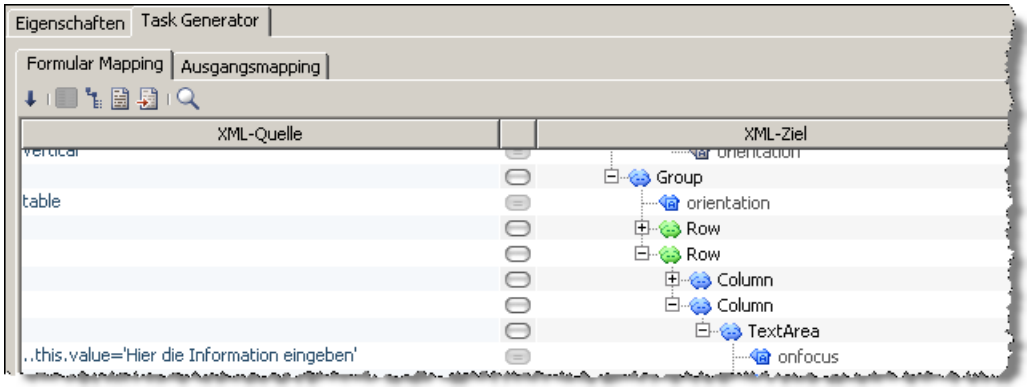
Nur beim Einsatz eines Formulars in einer Web Applikation haben Sie die Möglichkeit, den JavaScript-Code zentral im Web Application Connector im *Dialog „Interne Ressourcen“ (Workbench/Process Engine: Systemkonnektor-Guide, Kap. 39.3.2, S. 403)* einzugeben.

Vorteile: Sie können den JavaScript-Code

- ändern, ohne das XSLT-Stylesheet zu ändern,
- in allen Formularen des Technical Workflows verwenden.

Javascript-Eventhandler

JavaScript-Eventhandler fügen Sie direkt als Attribut des betroffenen Elements ein, z. B.



Das Element `TextArea` definiert ein Eingabefeld. Sobald der Mauszeiger in das Eingabefeld gestellt wird, wird der Eventhandler `onfocus` aktiv und zeigt in dem Eingabefeld die Aufforderung „Hier die Informationen eingeben“ an.

Informationen über die verschiedenen JavaScript-Eventhandler finden Sie unter http://de.selfhtml.org/javascript/sprache/eventhandler.htm.

4.2.10 Portlets und Formularabschnitte mit Ajax aktualisieren

Dieser Abschnitt erläutert die folgenden Themen:

- *Portlet mit Ajax aktualisieren, S. 77*
- *Formularabschnitt mit Ajax aktualisieren, S. 78*

Mit Ajax (Asynchronous JavaScript and XML) können Sie gezielt Inhalte einer Webseite austauschen, ohne dass die gesamte Webseite neu geladen werden muss. Die Inhalte werden vom Webserver geladen.

In der inubit Suite 6 können Sie Ajax in Web Applikationen nutzen, um gezielt auf einer Portalseite ein Portlet und Abschnitte in einem Portlet-Formular zu aktualisieren.

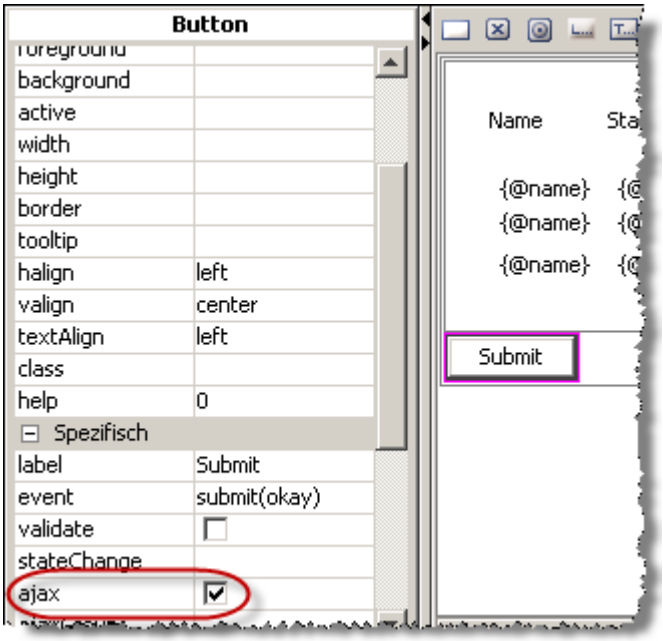
4.2.10.1 Portlet mit Ajax aktualisieren

Voraussetzungen

- Ajax-basiertes Aktualisieren ist nur verfügbar in Web Applikationen, die als Portlets in einen Portalserver deployt werden.
- Das Formular in Ihrer Web Applikation muss einen Button mit einem Submit-Event enthalten, weil die ajax-basierte Aktualisierung immer manuell über einen Klick auf solch einen Button angestoßen wird.

So gehen Sie vor

1. Zeigen Sie das Formular im Designer an.
2. Markieren Sie den Submit-Button.
   Im Register „Eigenschaften“ werden die Eigenschaften des Buttons angezeigt.
3. Aktivieren Sie die Option „ajax“:



4. Speichern Sie die Änderung und publizieren Sie den Task Generator.

Bei einem Klick auf den Ajax-Submit-Button im geänderten Portlet werden alle Formulardaten in dem Portlet aktualisiert. Falls weitere Portlets auf derselben Portalseite vorhanden sind, werden die Daten in diesen Portlets nicht neu geladen.

4.2.10.2 Formularabschnitt mit Ajax aktualisieren

Mit Ajax können Sie ausgewählte Inhalte eines Formulars auf einer Webseite ändern, ohne das gesamte Portlet neu zu laden.

Für Sie als Workflow-Designer hat dies folgende Vorteile:

- Sie müssen zum Neuaufbau der Seite nur die geänderten Daten im Task Generator erneut abbilden, nicht das gesamte Formular.
- Sie können flexibel auf aktuelle Benutzereingaben reagieren, indem Sie z. B. weitere Formularelemente passend zur letzten Eingabe einblenden
- Sie können Formulare übersichtlicher gestalten, indem Sie z. B. optionale Felder nur auf Bedarf einblenden.

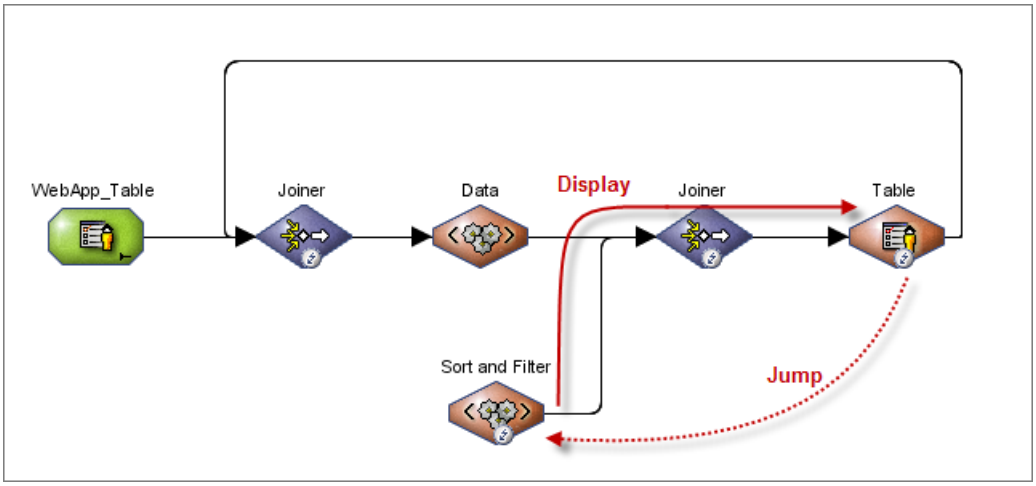
Dieser Abschnitt erläutert an einem Beispiel, wie Sie die Ajax-Aktualisierung konfigurieren. In dem Beispiel werden Tabellendaten sortiert.

Voraussetzungen

- Ajax-basiertes Aktualisieren ist nur verfügbar in Web Applikationen, die als Portlets in einen Portalserver deployt werden.
- Das Formular in Ihrer Web Applikation muss mind. einen Button mit einem Submit-Event enthalten, weil die ajax-basierte Aktualisierung immer manuell über einen Klick auf solch einen Button angestoßen wird.
- Der Workflow, der Ihre Web Applikation realisiert, enthält ein Modul, das die aktualisierten Daten bereit stellt. Dieses Modul ist über einen Pfad mit dem Task Generator verbunden.

Beispiel-Workflow

Der abgebildete Workflow erzeugt ein Portlet mit einer Tabelle. Die Tabellendaten können mit einem Klick auf- oder absteigend sortiert werden:



Das Sortieren ist mit Ajax realisiert: sobald Benutzer auf einen Spaltenkopf klicken, werden die aktuell angezeigten Daten vom Task Generator „Table“ zum Sortieren direkt an den XSLT Converter „Sort and Filter“ gesendet. In der Abbildung visualisiert die gestrichelte Linie „Jump“ diese Aktion.

Die Daten bestehen aus der Response und der Eingangsnachricht des Task Generators, d. h. aus genau den Daten, die normalerweise vom Browser an den Task Generator zurück gesendet werden, wenn Benutzer nach einer Formulareingabe den „Submit“-Button anklicken.

Der XSLT Converter sortiert die Daten und übergibt diese an das nächste Modul im Workflow. Der Workflow zwischen dem XSLT Converter und dem Task Generator wird, wie in der Abbildung, durch die durchgezogene Linie „Display“ signalisiert.

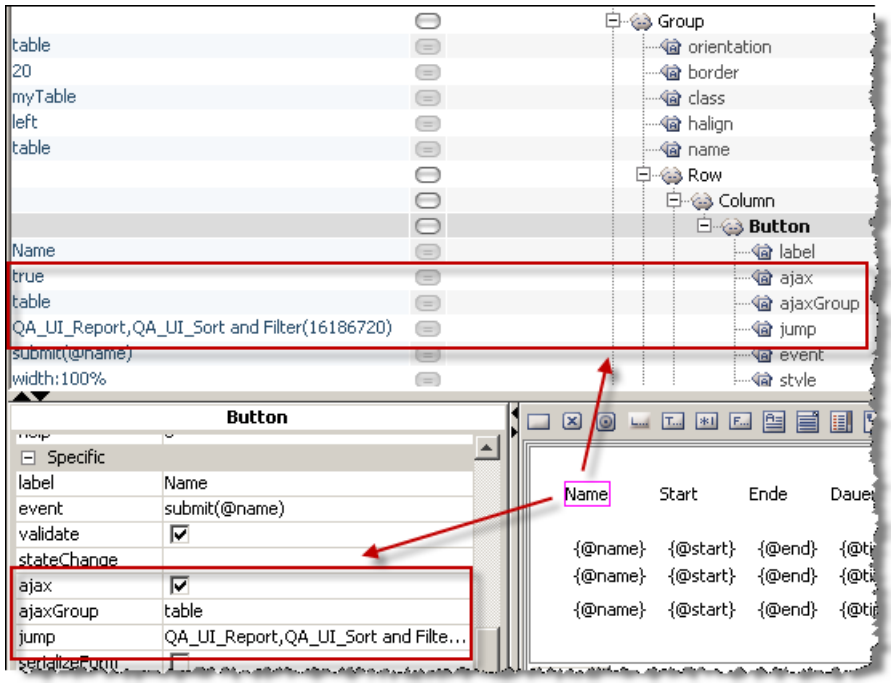
Der Task Generator erhält die aktualisierten Daten und führt das Formularmapping erneut durch. Deswegen erwartet der Task Generator die Daten in demselben Format wie die ursprüngliche Eingangsnachricht.

Stellen Sie sicher, dass die Daten in demselben Format wie die Eingangsnachricht vorliegen!

Statt eines einzelnen Moduls, wie dem XSLT Converter, kann natürlich auch ein komplexer Workflow zum Aktualisieren der Daten konfiguriert werden.

Konfiguration der Buttons

Die Spaltenköpfe sind als Submit-Buttons umgesetzt. Zusätzlich ist jeder Submit-Button folgendermaßen konfiguriert:



- **`ajax:`** Die Option ist aktiviert.
- **`ajaxGroup`**:
  Hier ist angegeben, welche Daten aktualisiert werden sollen. Der Wert bezieht sich immer auf das benanntes `group`-Element, welches die zu aktualisierenden Daten umschließen muss.
  Der Name des `Group`-Elements ist im `name`-Attribut des `Group`-Elements angegeben. In diesem Beispiel sollen alle Tabellendaten sortiert werden, daher lautet der Wert `table`.

Zwischen dem Element `<xsl:template match="/">` und dem benannten `Group`-Element dürfen folgende Befehle **nicht** angewendet werden: `call-template` und `apply-template`.

Stellen Sie sicher, dass das `group`-Element immer vorhanden ist! Das bedeutet u. a., dass das `group`-Element z. B. **nicht** durch ein `if`-Element bedingt werden darf.

- **`jump`**: Mit dieser Option wird das Modul festgelegt, an welches die Daten zur Aktualisierung gesendet werden.

4.3 Benutzereingaben in Formularen validieren

Dieser Abschnitt erläutert die folgenden Themen:

- *Clientseitige Validierung von Benutzereingaben über Pattern, S. 81*
- *Server-seitige Validierung von Benutzereingaben per AJAX, S. 82*

Zum Validieren der Benutzereingaben haben Sie folgende Möglichkeiten:

- **Client-seitige Validierung über Reguläre Ausdrücke**
  An den Formularelementen werden reguläre Ausdrücke und geeignete Fehlermeldung erfasst. Die Benutzereingaben werden vor dem Absenden des Formulars im Browser geprüft. Fehlerhafte Eingaben werden markiert, es wird eine Fehlermeldung ausgegeben und das Formular wird nicht abgeschickt.
- **Server-seitige, kontext-sensitive Validierung via AJAX**
  An den Formularelementen werden XPath-Ausdrücke und geeignete Fehlermeldung hinterlegt. Vor dem eigentlichen Absenden des Formulars wird ein Ajax-Call mit den Formularfeldern und Benutzereingaben an den Webserver gesendet. Dort werden die Daten anhand der XPath-Ausdrücken validiert. Im Browser werden die fehlerhaften Felder per Ajax markiert und eine Liste der Fehlermeldungen angezeigt.

4.3.1 Clientseitige Validierung von Benutzereingaben über Pattern

Um Benutzereingaben in Formularen noch vor dem Absenden im Browser prüfen zu lassen, können Sie für die Eingabefelder reguläre Ausdrücke definieren.

Bei einer fehlerhaften Eingabe werden die entsprechenden Felder markiert, es wird eine Fehlermeldung ausgegeben und das Formular wird nicht abgeschickt.

Voraussetzungen

Im Formular ist mind. ein Button-Element mit dem Event `submit` vorhanden. An diesem Element ist das Attribut `validate` markiert.

So gehen Sie vor

1. Fügen Sie im Designer ein `textField`-Element ein.
2. Markieren Sie das Element und zeigen Sie das Register „Eigenschaften“ an.
3. Formulieren Sie die Einschränkungen der Benutzereingabe als regulären Ausdruck. Geben Sie den Ausdruck als Wert des Attributs `pattern` ein.

Reguläre Ausdrücke müssen in Schrägstriche eingeschlossen werden, z. B. `/(\d\d).(\d\d).(\d\d\d\d)/`.

4. Geben Sie die Fehlermeldung, die angezeigt werden soll, wenn die Benutzereingabe nicht dem definierten Pattern entspricht, als Wert des Attributs `errorText` ein.
5. Wenn das Formular nicht im inubit Enterprise Portal angezeigt werden soll, dann müssen Sie das Element `<textArea name='errorViewer'>` für die Anzeige der Fehlermeldung hinzufügen. Das Element muss hierarchisch über dem Eingabefeld stehen, an dem `pattern` definiert ist. Es ist standardmäßig ausgeblendet.
6. Publizieren Sie das Modul.

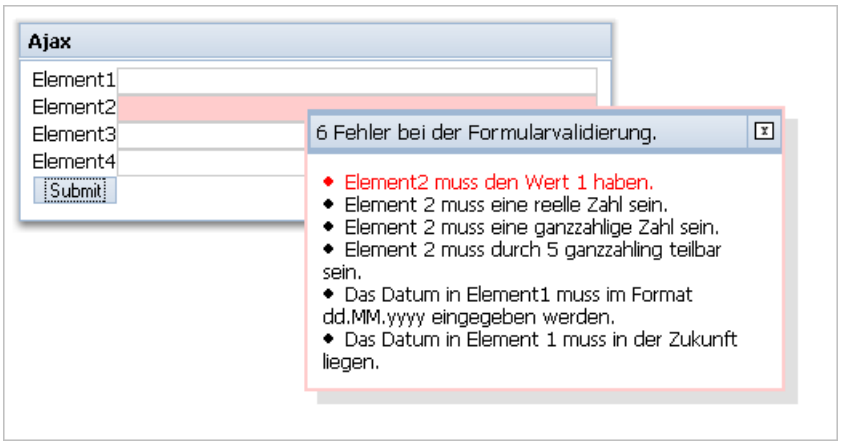
4.3.2 Server-seitige Validierung von Benutzereingaben per AJAX

Die Benutzereingaben in Formulare können kontextsensitiv mit Hilfe selbstformulierter Validierungsregeln überprüft werden. Folgende Prüfungen sind z. B. möglich:

- Obligatorische Felder gefüllt?
- Datentypen der Eingabewerte korrekt?
- Datumsformate korrekt?

Die Benutzereingaben werden server-seitig validiert, dazu werden die Benutzereingaben im Hintergrund an die inubit Process Engine gesendet. Ausgelöst wird die Validierung durch ein Submit-Event, die Benutzereingaben werden also validiert, sobald ein Submit- bzw. Senden-Button im Formular angeklickt wird.

Die Ergebnisse der Validierung werden im Formular als Liste in einem Tooltipp angezeigt. Ein Klick auf ein einzelnes Validierungsergebnis markiert das Eingabefeld, in welches der ungültige Wert eingegeben wurde:


Ajax
Element1
Element2
Element3
Element4
Submit
6 Fehler bei der Formularvalidierung.
Element2 muss den Wert 1 haben.
Element 2 muss eine reelle Zahl sein.
Element 2 muss eine ganzzahlige Zahl sein.
Element 2 muss durch 5 ganzzahling teilbar sein.
Das Datum in Element1 muss im Format dd.MM.yyyy eingegeben werden.
Das Datum in Element 1 muss in der Zukunft liegen.

Technischer Hintergrund

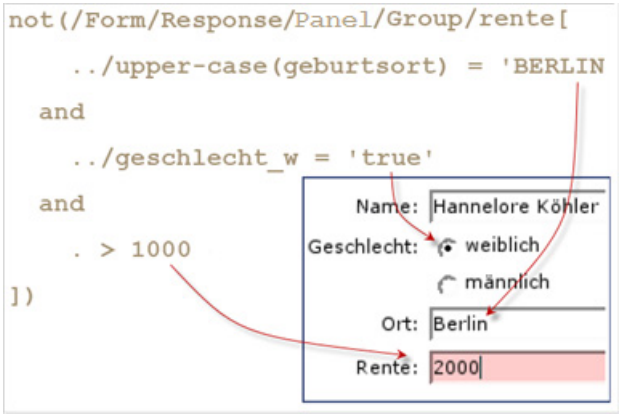
Die Validierungsregeln werden im Standard von WS-Policy MessagePredicates erfasst, d. h. in Form von XPath-Ausdrücken. Die gesamte XPath 2.0-Syntax wird unterstützt.

Alle Validierungsregeln beziehen sich direkt auf die Formularstruktur und sind wiederverwendbar. Sie können beliebig komplexe Abhängigkeiten zwischen Formularfeldern abbilden.

Beispiel

In einem Formular werden die Rentenhöhe, der Wohnort und das Geschlecht von Rentnern abgefragt. Die Validierungsregel lautet umgangssprachlich: „Die monatliche Rente einer Berlinerin darf nicht über 1000 Euro liegen“.

Als XPath-Ausdruck sieht die Validierungsregel so aus:



Voraussetzungen

- Im Formular ist mind. ein Button-Element mit dem Event `submit` vorhanden.
- Bei dem Button-Element ist das Attribut `validate` markiert.
- Sie verfügen über XPath-Kenntnisse.

So gehen Sie vor

1. Zeigen Sie das Formular im Formular-Designer an.
2. Wählen Sie „Bearbeiten > Server-seitige Validierung…“.
   Der Editor öffnet sich und zeigt ein XML-Dokument mit Beispielen für Validierungsregeln an.
3. Passen Sie die Beispiele an Ihre Formularstrukturen an oder fügen Sie neue Elemente hinzu.
   Damit bei einem Klick auf ein Validierungsergebnis immer das dazugehörige Eingabefeld markiert wird, müssen die Validierungsregeln nach folgendem Muster erstellt werden:
   `not(invalidesElement).`

Beispiel: Eine Regel lautet umgangssprachlich „Das Eingabefeld „InputElement“ darf nicht leer sein“, als Pseudo-Code „nicht(Eingabefeld InputElement leer)“. Der korrekte XPath-Ausdruck lautet „`not(/Form/Response/Panel/Group/InputElement[string-length(.)=0])`“

Standardmäßig können Validierungsergebnisse auf Deutsch und Englisch angezeigt werden. Um weitere Sprachen zu ergänzen, fügen Sie pro Sprache folgendes ein:
- ein wsp:PolicyAttachment-Attribut, z. B.
`inubit:heading.es='{errorCount} Error durante la validación del formulario'.`
Diese Angabe wird als Titel der Ergebnisliste verwendet.
- ein HTML-List-Element mit einer Erklärung, z. B.
`inubit:responseHtml.es='<li>Elemento1 es un campo obligatorio.</li>'`

4. Um Ihre Änderungen zu speichern, klicken Sie auf „OK“.

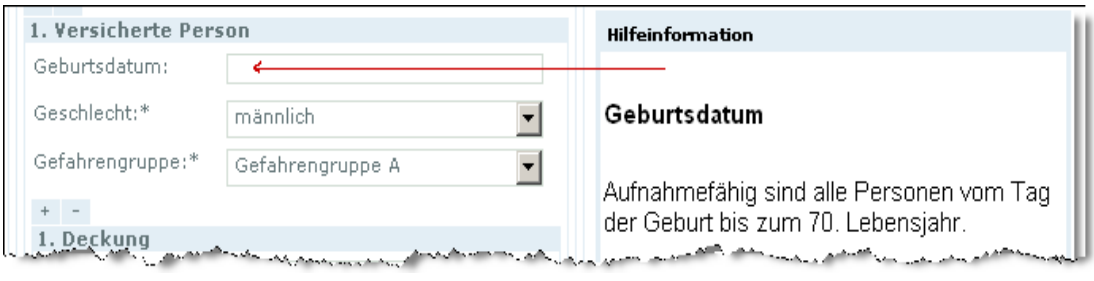
Ihre Änderungen werden gespeichert und der Editor schließt sich.

4.4 Kontextsensitive Onlinehilfe erstellen

Sie können für Formular-Elemente Hilfe-Texte erstellen und situationsabhängig anzeigen lassen. Für jedes Formular-Element können ein oder mehrere Hilfe-Texte vorhanden sein.

Beispiel

Sobald Benutzer in das Feld „Geburtsdatum“ klicken, wird ein Hilfe-Text mit Informationen über das Alter als Versicherungskriterium angezeigt:



Technischer Hintergrund

Die Anzeige des Hilfe-Textes für das Eingabefeld „Geburtsdatum“ wird ausgelöst, sobald das Eingabefeld den Fokus erhält. Dies wird in einer XML-Datei mit Hilfe eines XPath-Ausdrucks folgendermaßen ausgedrückt:

```
</inubit:HelpPredicate>
  <inubit:HelpPredicate inubit:disabled="false"
    inubit:elementName="gebDatum">
    local-name(.)='gebDatum'
```

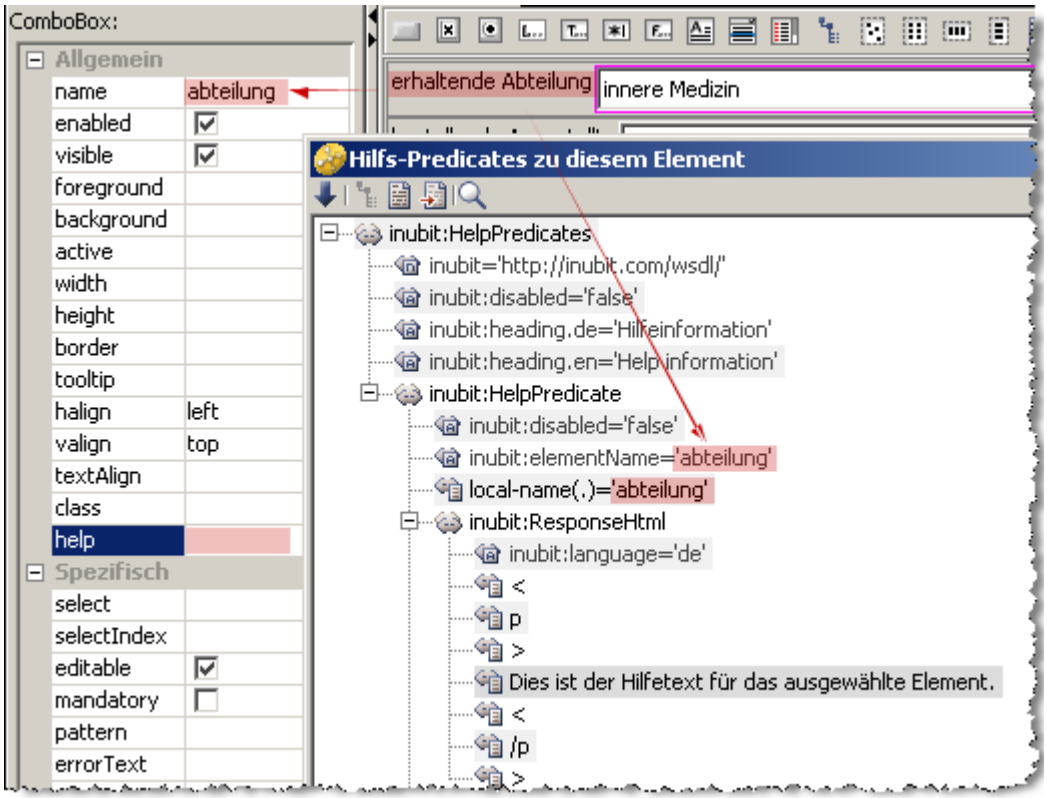
Die XML-Struktur und der Feldname werden automatisch beim Definieren des Hilfe-Textes korrekt erzeugt.

Danach folgt der eigentliche Hilfe-Text:

```
    <inubit:ResponseHtml inubit:language="de">
      Aufnahmefähig sind alle Personen vom Tag
      der Geburt bis zum 70. Lebensjahr.
    </inubit:ResponseHtml>
</inubit:HelpPredicate>
```

So gehen Sie vor

1. Zeigen Sie das Formular im Formular Designer an.
2. Markieren Sie das Formular-Element, zu dem Sie einen Hilfe-Text erstellen möchten, und zeigen Sie das Register „Eigenschaften“ an.
3. Klicken Sie im Bereich „Allgemein“ rechts neben dem Attribut „help“ auf das leere Feld. Der Editor öffnet sich und zeigt ein XML-Dokument mit einem so genannten „HelpPredicate“ für das ausgewählte Formular-Element an, z. B.:



4. Ersetzen Sie den Inhalt des Elementes `inubit:ResponseHTML` durch Ihren Hilfetext.

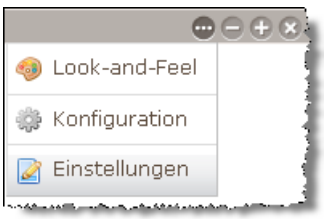
Mehrsprachigkeit:
Standardmäßig werden Hilfe-Texte auf Deutsch und Englisch angezeigt. So zeigen Sie z. B. spanische Hilfe-Texte an:
- Fügen Sie in das Element `inubit:HelpPredicates` folgendes Attribut ein: `heading.es='información on-line'` Der Text wird als Titel des Hilfe-Texts angezeigt.
- Umschließen Sie jeden spanischen Hilfe-Text mit dem Element `<inubit:responseHtml inubit:language.es>'`

5. Zum Speichern klicken Sie auf „OK“.
Der Editor schließt sich und Ihre Änderungen werden gespeichert. Sobald Sie einen Hilfe-Text eingegeben haben, beträgt der Wert des `help`-Attributs 1. Wenn mehrere Hilfe-Texte vorhanden sind, wird die Anzahl entsprechend hochgezählt.

Um alle Hilfetexte eines Formulars anzuzeigen oder zu bearbeiten, wählen Sie „Bearbeiten > Alle Hilfetexte…“. Der Editor öffnet sich und zeigt ein XML-Dokument mit allen Hilfe-Texten an.

6. Aktivieren Sie nun die Anzeige der Hilfe im Portal:
   a. Zeigen Sie mit dem Cursor auf den rechten oberen Rand der Tasklist bzw. des Portlets, in dem Ihr Formular angezeigt wird. Eine Reihe von Buttons wird eingeblendet.

b. Klicken Sie auf den Button „Einstellungen“:



Die allgemeinen Einstellungen werden angezeigt:

c. Klicken Sie auf „Hilfe aktivieren“.

Die Tasklist bzw. Ihr Portlet sowie Ihre Hilfetexte werden angezeigt.

4.5 Formular-Texte zum Übersetzen exportieren und importieren

Damit Sie Formular-Texte in eine oder mehrere Sprachen übersetzen lassen können, bietet die inubit Suite 6 eine Export-/Import-Funktion über eine Excel-Datei an.

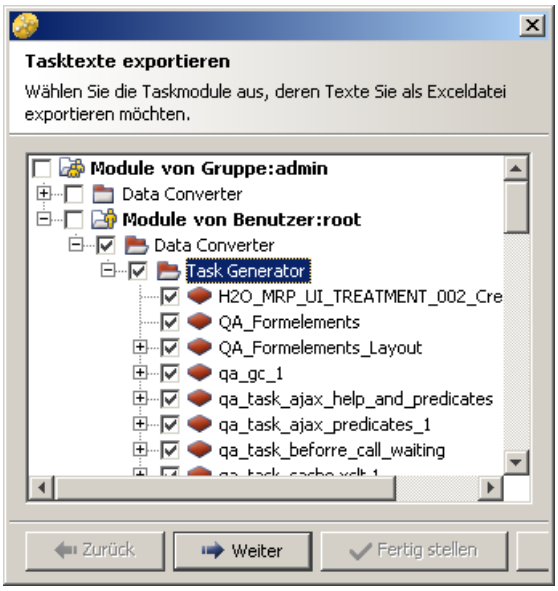
Alle Bezeichner von Formular-Elementen (`label`-Attribute und deren Werte) werden in eine Excel-Datei exportiert. Für jede Zielsprache wird eine Tabellenspalte für die Übersetzungstexte erstellt.

Nach der Übersetzung können Sie die Datei wieder in die inubit Suite 6 importieren.

Export

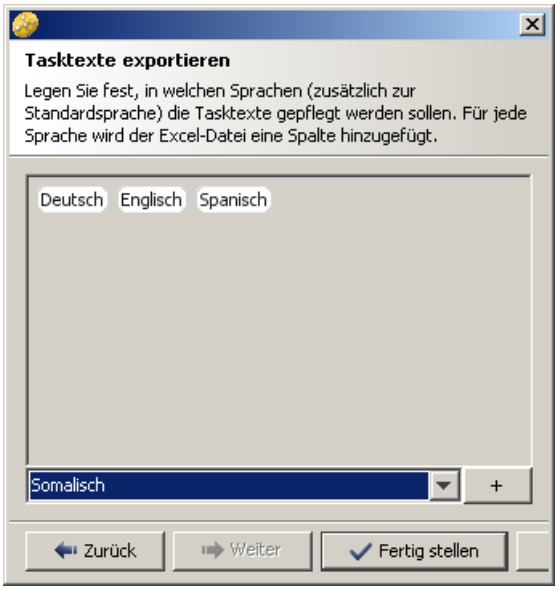
So gehen Sie vor

1. Zeigen Sie in der inubit Workbench das Register „Modul-Editor > Server“ an.
2. Wählen Sie in der Menüleiste „Bearbeiten > Exportieren > Tasktexte“. Der folgende Dialog wird angezeigt:



Es werden alle Task Generatoren auf der inubit Process Engine aufgelistet, die Sie (entsprechend Ihrer Rolle) anzeigen/bearbeiten können.

3. Markieren Sie die Module, von denen Sie die Texte exportieren möchten.
4. Klicken Sie auf „Weiter“. Der folgende Dialog öffnet sich:



5. Geben Sie an, für welche Sprachen Übersetzungen erstellt werden sollen. Für jede Sprache wird in der Excel-Exportdatei eine Tabellenspalte mit dem Namen `new.label.[Länderkürzel]` erstellt. Wenn die Label bereits übersetzt worden sind, wird eine zusätzliche Spalte `old.label.[Länderkürzel]` mit der bisherigen Übersetzung erzeugt:
   a. Wählen Sie aus der DropDown-Liste eine Sprache aus.
   b. Klicken Sie auf den Plus-Button. Die Sprache wird im oberen Bereich angezeigt.

c. Wiederholen Sie diese Schritte für alle Sprachen.

6. Klicken Sie auf „Fertig stellen“. Eine Excel-Datei mit den vorhandenen Labeln wird erstellt und ein Dateiexplorer angezeigt.
7. Geben Sie den Speicherort und evtl. den Dateinamen an und speichern Sie die Excel-Datei.

In der Excel-Datei sind die Ausgangstexte und alle Ziel-Sprachen jeweils in einer eigenen Tabellenspalte gespeichert. Die Spalte mit den Ausgangstexten ist gelb hinterlegt, die Spalten mit zu übersetzenden Texten sind weiß hinterlegt.

Ändern Sie ausschließlich Texte in weißen Tabellenspalten; gelb hinterlegte Spalten enthalten Original-Texte bzw. frühere Übersetzungen und dürfen nicht bearbeitet werden!
Bearbeiten Sie die Formulare nicht, solange die Task-Texte zum Übersetzen exportiert sind. Wenn die exportierten und übersetzten Texte beim Import nicht mit den vorhandenen Texten im Modul übereinstimmen, können Sie die Texte nicht wieder importieren.

Import

So gehen Sie vor

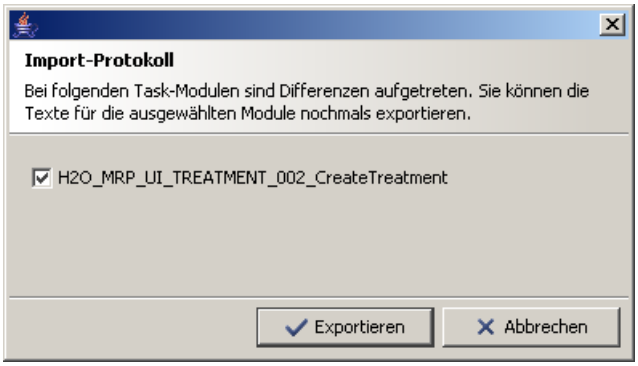
1. Zeigen Sie im Module-Editor der inubit Workbench das Register „Server“ an.
2. Wählen Sie in der Menüleiste „Bearbeiten > Importieren > Tasktexte“. Ein Dateiexplorer wird angezeigt.
3. Navigieren Sie zu der Excel-Datei mit den Übersetzungen und öffnen Sie diese im Dateiexplorer.

Nachdem die Task-Texte erfolgreich importiert wurden, erhalten Sie eine entsprechende Meldung.

Problembehebung beim Import

Bei Problemen mit dem Import werden

- trotzdem alle Texte importiert, die eindeutig zugeordnet werden können,
- ein Import-Protokoll angezeigt, in dem die Module aufgelistet werden, bei denen Differenzen auftraten, z. B.


Import-Protokoll
Bei folgenden Task-Modulen sind Differenzen aufgetreten. Sie können die Texte für die ausgewählten Module nochmals exportieren.
H2O_MRP_UI_TREATMENT_002_CreateTreatment
Exportieren
Abbrechen

Sie haben die Möglichkeit, die Texte für das angezeigte Modul erneut zu exportieren, um die aktuellen Task-Texte und deren Übersetzungen zu überprüfen.

4.6 Widgets verwenden und erstellen

Dieser Abschnitt erläutert die folgenden Themen:

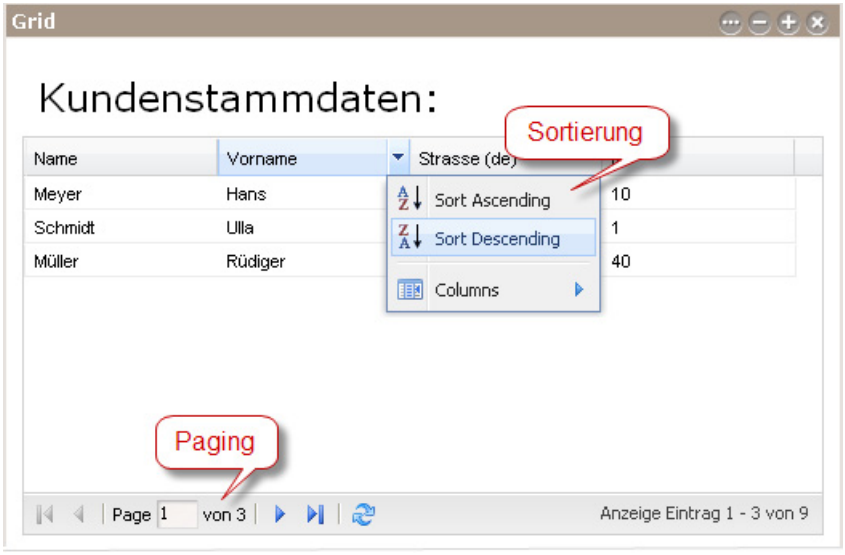
- *Grid-Widget verwenden und konfigurieren, S. 91*
- *Widget erstellen und in die inubit Suite 6 integrieren, S. 97*

Widgets sind wiederverwendbare Steuerelemente für grafische Benutzeroberflächen von Web-Applikationen, wie z. B. Kalender, Toolbars, Menüs, RSS-Feeds oder Tabellen.

Das Widget Framework der inubit Suite 6 bietet Ihnen die Möglichkeit, eigene Widgets zu erstellen. Für Ihre Widgets können Sie z. B. die Funktionen des Ext JS-Frameworks nutzen, das zusammen mit der iS ausgeliefert wird. Das Ext JS-Framework ist ein client-seitiges JavaScript- bzw. Ajax-Framework und bietet eine umfangreiche Sammlung von Steuerelementen.

→ Siehe *Widget erstellen und in die inubit Suite 6 integrieren (Workbench/Process Engine: Modul-Guide, Kap. 4.6.2, S. 97).*

Im Lieferumfang der inubit Suite 6 ist das einsatzbereite Widget „Grid“ enthalten. Dieses Widget stellt Daten als Tabelle dar und bietet u. a. Paging und Sortierung auf Basis des Ext JS-Grids, z. B.:



→ Siehe *Grid-Widget verwenden und konfigurieren (Workbench/ Process Engine: Modul-Guide, Kap. 4.6.1, S. 91).*

Funktionsprinzip

Ein Widget besteht aus HTML-, JavaScript- und CSS-Code. Im iS wird dieser Code bei der Ausführung des Task Generators genauso dynamisch erzeugt wie das „normale" Formular, in dem das Widget eingebettet ist:

- Die XML-Struktur des Formulars, die im Register „Formular Mapping" des Task Generators angezeigt wird, enthält auch die XML-Struktur des Widgets. Für das mitgelieferte Grid-Widget wird z. B eine XML-Tabellenstruktur aufgebaut, die zusätzlich Daten zur Konfiguration der Tabelle enthält, wie z. B. für das Paging, die maximale Anzahl der Datensätze auf einer Seite.

  Meist wird diese XML-Struktur dynamisch mit Daten aus der Eingangsnachricht gefüllt.
- Ein widget-spezifisches XSLT-Stylesheet erzeugt aus der XML-Struktur den HTML-Code, der im Browser im Formular angezeigt wird. Das widget-spezifische XSLT-Stylesheet enthält neben XSLT-Befehlen auch JavaScript- und CSS-Code, mit dem Funktionalität und Layout des Widgets realisiert werden.

Die XML-Struktur und das XSLT-Stylesheet des Widgets werden im Repository abgelegt. Damit bleibt die XML-Struktur des Formulars im Task Generator kompakt und lesbar und das Widget kann in anderen Task Generatoren wiederverwendet werden.

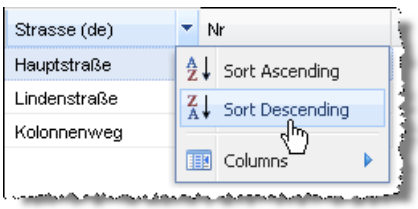
Widgets werden, genau wie die anderen GUI-Komponenten, per Drag'n'Drop in ein Formular integriert.

4.6.1 Grid-Widget verwenden und konfigurieren

Mit dem Grid-Widget können Sie in Ihrer Web-Applikationen Daten in Tabellenform darstellen. Das Grid-Widget basiert auf Funktionen des Ext JS-Frameworks und bietet folgende Features:

- **Sortierung**

  Spalteninhalt auf- oder absteigend sortieren:



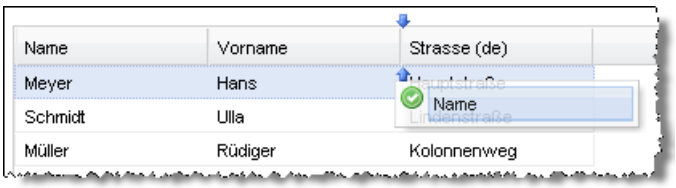
  Abhängig von der Konfiguration der Tabelle findet die Sortierung server- oder client-seitig statt.

  → Siehe

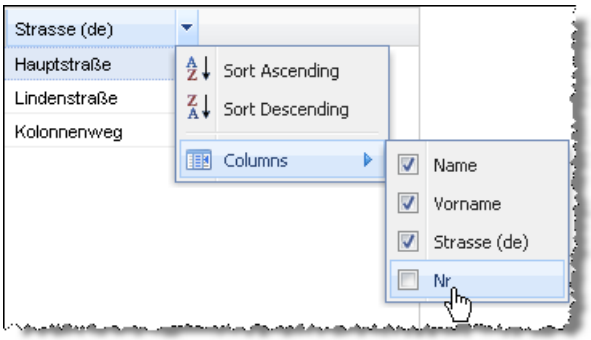
  - *Beispiel: Dynamische Tabelle mit client-seitiger Sortierung, ohne Paging (Workbench/Process Engine: Modul-Guide, Kap. 4.6.1.1, S. 93)*

- *Beispiel: Dynamische Tabelle mit Paging und server-seitiger Sortierung (Workbench/Process Engine: Modul-Guide, Kap. 4.6.1.2, S. 94)*

- **Spaltenreihenfolge ändern**
  Per Drag'n'Drop:



- **Spalten ein/ausblenden**

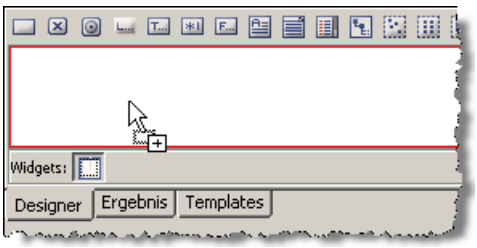


- **Paging**
  Große Datenmengen können aufgeteilt und seitenweise angezeigt werden. Die Anzahl der auf einer Seite angezeigten Daten ist konfigurierbar, die Daten werden dynamisch über Ajax-Calls nachgeladen.

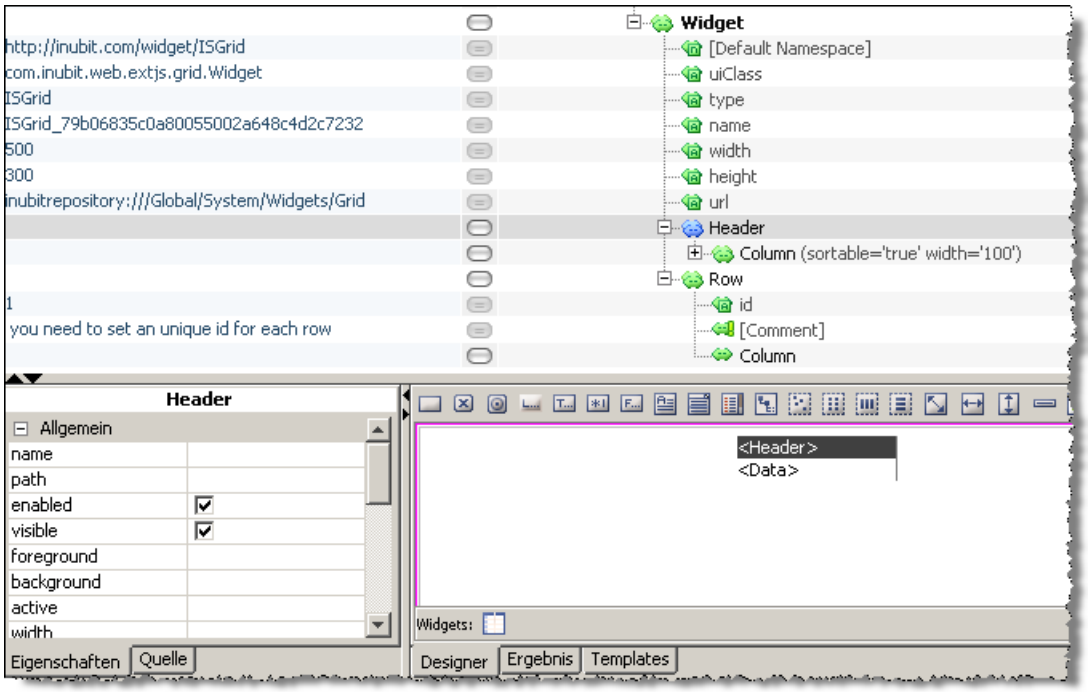
→ Siehe *Beispiel: Dynamische Tabelle mit client-seitiger Sortierung, ohne Paging (Workbench/Process Engine: Modul-Guide, Kap. 4.6.1.1, S. 93).*

So gehen Sie vor

1. Zeigen Sie im Task Generator das Register „Formular Mapping" an.
2. Ziehen Sie das Grid-Icon aus der Widget-Leiste des Designers in dessen Arbeitsfläche:



Im Bereich „XML-Ziel" wird die erwartete XML-Struktur des Grids angezeigt:

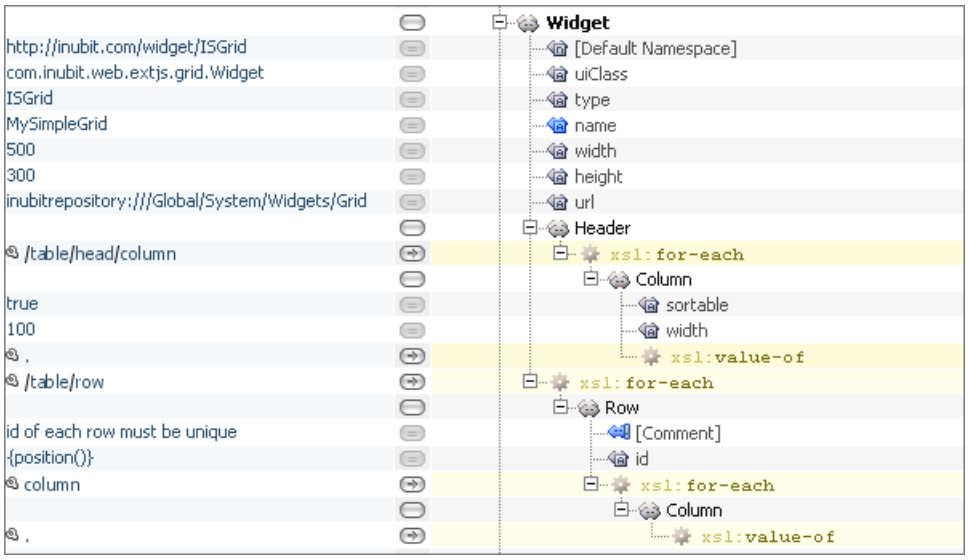


Sie können die XML-Struktur des Grids statisch oder dynamisch mit Daten füllen. Beispiele finden Sie in den folgenden Abschnitten:

- *Beispiel: Dynamische Tabelle mit client-seitiger Sortierung, ohne Paging (Workbench/Process Engine: Modul-Guide, Kap. 4.6.1.1, S. 93)*
- *Beispiel: Dynamische Tabelle mit Paging und server-seitiger Sortierung (Workbench/Process Engine: Modul-Guide, Kap. 4.6.1.2, S. 94)*

4.6.1.1 Beispiel: Dynamische Tabelle mit client-seitiger Sortierung, ohne Paging

Die folgende Abbildung zeigt das Formular Mapping einer Tabelle, die durch XML-Eingangsnachrichten befüllt wird. Die Anzahl der Reihen und Spalten wird durch die XML-Eingangsnachrichten definiert:

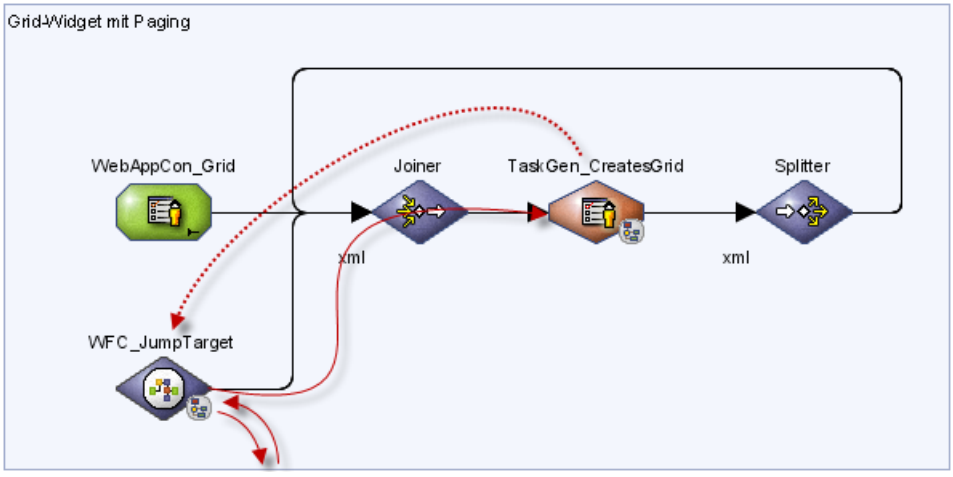


Die Sortierung nach Tabellenspalten steht automatisch zur Verfügung, solange das `sortable`-Attribut des `Column`-Elements den Wert `true` hat.

4.6.1.2 Beispiel: Dynamische Tabelle mit Paging und server-seitiger Sortierung

Der Einsatz des Paging-Mechanismus und der server-seitigen Sortierung setzt voraus, dass die Daten im Grid-Widget aus einer Datenquelle geladen werden können.

Dazu muss der Workflow, der Ihr Grid-Widget als Portlet realisiert, wie in der folgenden Abbildung aufgebaut sein:



Der Task Generator, welcher das Grid-Widget erzeugt, ist über einen Pfad mit der Datenquelle verbunden: In dem abgebildeten Workflow beginnt dieser Pfad bei dem Workflow Connector `WFC_JumpTarget` und führt zu einem Database Connector in einem Subworkflow, von

dort aus wieder zurück zum Task Generator. Die Einsprungstelle in diesen Pfad, also die ID des ersten Moduls, wird im Grid-Widget eingegeben.

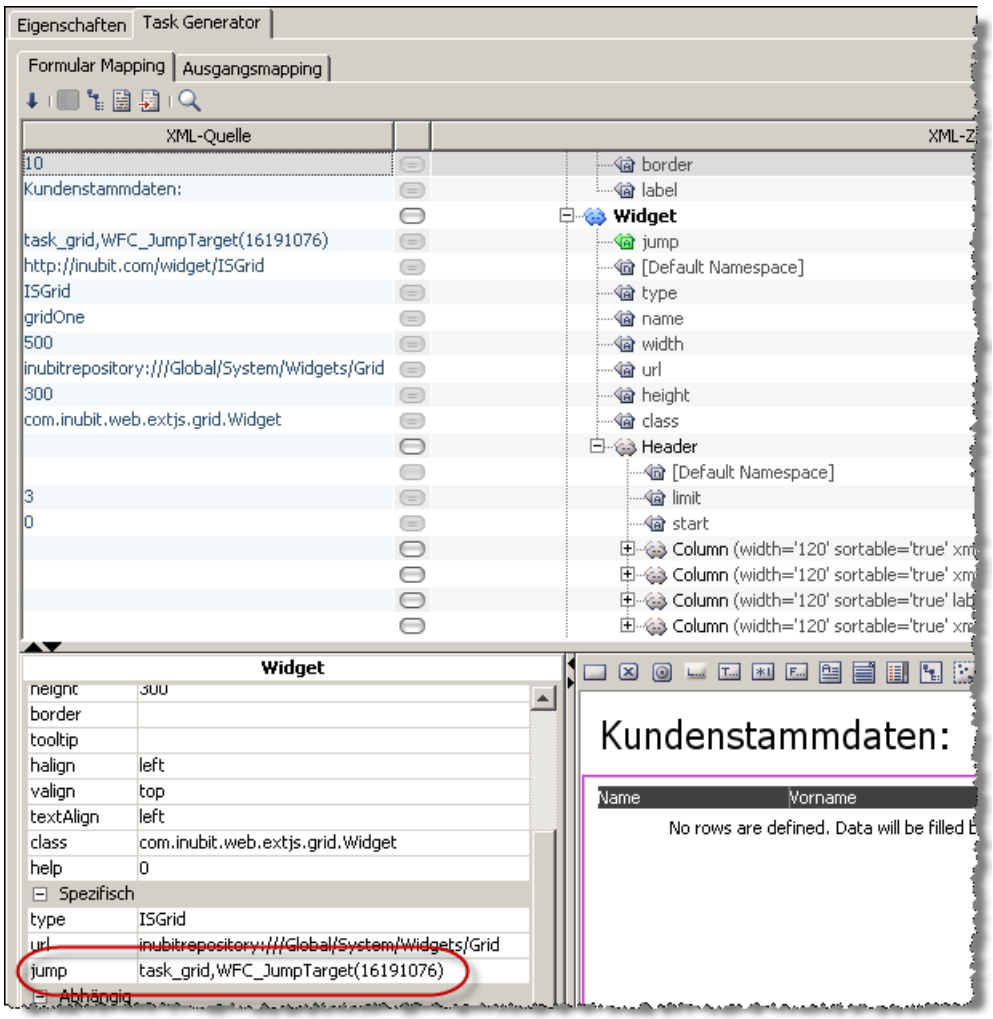
Benutzeraktionen, wie z. B. die Anzeige des Grid-Portlets oder das Sortieren der Daten, lösen bei dem Task Generator einen Request aus. Der Inhalt des Requests ist abhängig von der Benutzeraktion: Bei einer Sortieraktion enthält der Request z. B. folgende Informationen:

- Erster Datensatz, der auf der Seite angezeigt werden sollen
- Anzahl aller Datensätze, die auf der Seite angezeigt werden sollen
- Gewähltes Sortierkriterium (aufsteigend/absteigend)

Der Request wird an das erste Modul im Pfad übergeben und von diesem weitergeleitet. Der Database Connector wertet den Request aus und stellt die Daten bereit. Danach werden die angeforderten Daten an den Task Generator gesendet und in der Tabelle angezeigt.

Einsprungstelle in Pfad zur Datenquelle festlegen

Die folgende Abbildung zeigt das Formular Mapping der Tabelle:



Das erste Modul im Pfad zur Datenquelle definieren Sie im Register „Eigenschaften" über das `jump`-Attribut des `Widget`-Elements: Ein Klick auf die leere Zelle neben dem Attribut öffnet einen Dialog, in dem Sie das Modul auswählen.

Das Modul muss in demselben Workflow wie der Task Generator liegen!

Ob und welche Module zwischen der Einsprungstelle und der Datenquelle liegen, ist abhängig davon, ob der Request weiter aufbereitetet werden muss. Dies kann z. B. nötig sein, wenn Sie die Daten server-seitig sortieren lassen möchten.

→ Siehe *Server-seitige Sortierung konfigurieren (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 96)*.

Anzahl der angezeigten Datensätze festlegen

Die Anzahl der Datensätze pro Seite legen Sie im Register „Eigenschaften" über die `start`- und `limit`-Attribute des `Header`-Elements fest:

- `start`: Nummer des ersten angezeigten Datensatzes
- `limit`: Anzahl der angezeigten Datensätze

Server-seitige Sortierung konfigurieren

Die Sortierung müssen Sie manuell implementieren. Wenn Sie als Datenquelle eine Datenbank einsetzen, dann können Sie die Sortierkriterien z. B. durch entsprechende SQL-Befehle umsetzen.

Die Sortierkriterien sind in dem Request enthalten, der an die Datenquelle gesendet wird. Sie können den Request anzeigen, in dem Sie im Watch-Modus den Watchpoint vor der Datenquelle öffnen.

4.6.2 Widget erstellen und in die inubit Suite 6 integrieren

Benötigte Dateien

Um ein eigenes Widget zu entwickeln, müssen Sie folgende Dateien erstellen:

Name	Funktion
`Widget.xml`	XML-Struktur des Widgets. Diese hat zwei Funktionen: 1. Bildet die Struktur, in der Daten aus der Eingangsnachricht angezeigt werden. Bei dem mitgelieferten Grid-Widget wird z. B. eine Tabellenstruktur aufgebaut. 2. Dient der Konfiguration des Widgets: In der XML-Struktur können alle Parameter definiert werden, die z. B. der JavaScript-Code des Widgets benötigt. Die Parameterwerte können im Designer im Register „Eigenschaften" angegeben werden, genauso wie bei den anderen GUI-Komponenten im Designer. Beim mitgelieferten Grid-Widget wird z. B. die Anzahl der maximal sichtbaren Datensätze pro Seite über die `start` und `limit`-Attribute des Header-Elements definiert. → Siehe *`Widget.xml` (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 99)*.
`Widget.xsl`	XSLT-Stylesheet für die Darstellung des Widgets in der Web-Applikation. Enthält z. B. HTML, JavaScript-Code (evtl. Ext JS-Framework-Code) für die Funktionalität des Widgets und Layout-Informationen (als CSS-Referenzen). → Siehe *`Widget.xsl` (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 99)*.
`Widget.gif`	Grafik für das Widget-Icon in der Symbolleiste des Task Generators. Über das Icon kann die XML-Struktur per Drag'n'Drop an der gewünschten Position im Formular eingefügt werden. Die Grafik muss im gif-Format mit einer max. Höhe von 14 Pixel vorliegen.
`Widget.jar`	Optional. Java-Klasse für die Darstellung des Widgets im Reiter „Formular Mapping“ des Task Generators. Die Klasse muss `com.inubit.ibis.configuration.form.XulWidget` implementieren. Wenn Sie keine Java-Klasse bereit stellen, müssen Sie in der XML-Struktur Ihre Widgets (`Widget.xml`) bei dem Element `wd:Description` das Attribut `uiClass` entfernen. Wenn keine Java-Klasse vorhanden ist, wird im Designer ein Platzhalter statt des Widgets angezeigt. Die Funktionalität des Widgets wird dadurch nicht beeinträchtig.

Die Dateien müssen im globalen Bereich des Repositorys unter „Global > System > Widgets“ in einem eigenen Ordner verfügbar gemacht werden und anonym lesbar sein.

Siehe

- Ext JS API-Dokumentation: http://<NameIhrerinubit Process Engine>:8000/html/js/extjs/docs
- Ext JS Beispiele: http://<NameIhrerinubit Process Engine>:8000/html/js/extjs/examples/samples.html
- Ext JS Homepage: http://extjs.com

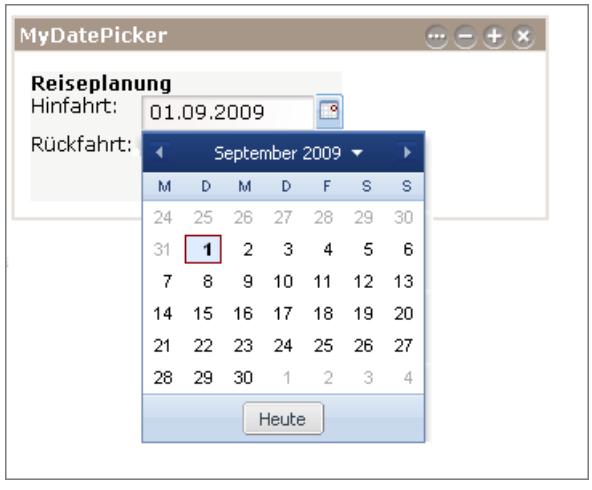
- ExtJS Starters-Page: http://easyext.co.uk

So gehen Sie vor

1. Nutzen Sie die Dateien `Widget.xml` und `Widget.xsl` des mitgelieferten Grid-Widgets als Vorlagen: Downloaden Sie diese Dateien aus dem Repository-Verzeichnis „Global > System > Widgets > Grid“.
2. Öffnen Sie die Datei `Widget.xml` und passen Sie die XML-Struktur an. Beachten Sie dabei die Kommentare in der Datei, diese erläutern die Verwendung der verschiedenen XML-Elemente.
3. Passen Sie die Datei `Widget.xsl` an. Beachten Sie auch in dieser Datei die Kommentare.
4. Erzeugen Sie ein Icon mit dem Namen `Widget.gif`.
5. Erzeugen Sie im Repository ein neues Verzeichnis unterhalb von „Global > System> Widgets“. Fügen Sie alle Dateien diesem Verzeichnis hinzu. Stellen Sie sicher, dass die Dateien anonym lesbar sind.
6. Wenn Sie das neue Widget sofort einsetzen möchten, müssen Sie das Icon zuerst sichtbar machen: Öffnen Sie einen Task Generator, zeigen Sie das Register „Formular Mapping" an und wählen Sie aus der Menüleiste „Ansicht > Externe Elemente neu laden“.
   Ansonsten steht das Widget erst nach dem Neustart der inubit Workbench zur Verfügung.

Beispiel: Datepicker-Widget

Ein Datepicker vereinfacht die Eingabe eines Datums:



Der Datepicker nutzt das Ext JS-Framework und basiert auf der Klasse `DateField`.

Siehe http://extjs.com/deploy/dev/docs/?class=Ext.form.NumberField

- **Widget.xml**

Listing

```
?xml version="1.0" encoding="UTF-8"?>
<wd:Description xmlns:wd="http://inubit.com/widget/schema"
type="ExtDateField" uiClass="com.inubit.web.extjs.form.ExtDateFieldWidget"
namespace="http://inubit.com/widget/ExtForm" prefix="">
  <Widget>
    <wd:Attributes>
      <!-- Beachten Sie, wenn Sie mehrere Datepicker in einem Formular
verwenden, dass jede Instanz einen eindeutigen Namen haben muss -->
      <name/>
    </wd:Attributes>
    <DateField>
      <wd:Attributes>
      <!-- Das Datumsfeld soll 300 px breit sein; das TextField kann alle
Attribute eines "normalen" TextFields erhalten -->
        <width>300</width>
      </wd:Attributes>
    </DateField>
    <Meta>
      <wd:Attributes>
      <!-- Der format-Wert kann im Designer definiert werden und legt fest,
in welchem Format das Datum angezeigt wird; zulässige Werte sind vom
Ext JS-Framework vorgegeben -->
        <format/>
      </wd:Attributes>
    </Meta>
  </Widget>
</wd:Description>
```

/Listing

- **Widget.xsl**

Listing

```
<?xml version="1.0" encoding="UTF-8"?>
  <xsl:output method="html" encoding="UTF-8"/>
  <xsl:template match="extform:Widget[@type='ExtDateField']">
    <xsl:variable name="wid" select="concat($ns,@name)"/>
    <xsl:variable name="tf">
      <TextField name="{@name}" xmlns="">
        <xsl:for-each select="extform:DateField/
@*[name()!='name']"><xsl:attribute name="{local-name(.)}"><xsl:value-of
select="."/></xsl:attribute></xsl:for-each>
        <xsl:value-of select="extform:DateField"/>
      </TextField>
```

/Listing (Abschnitt 1 von 2)

Listing

```
    </xsl:variable>
    <script language="javascript" type="text/javascript">
      addCssIS('/html/js/extjs/resources/css/ext-all.css');
      addCssIS('/html/js/extjs/resources/css/xtheme-gray.css');
    </script>
    <xsl:apply-templates select="$tf/TextField"/>
    <script language="javascript" type="text/javascript">
    Ext.onReady(function(){
      try {
        new Ext.form.DateField({
          applyTo: '<xsl:value-of select="$wid"/>'
          <xsl:if test="extform:DefaultTextField/@mandatory = 'true'">
            ,allowBlank: false
          </xsl:if>
          <xsl:if test="extform:Meta/@format != ''">
             ,format: '<xsl:value-of select="extform:Meta/@format"/>'
          </xsl:if>
        });
        }
        catch(e) {
          showMessageIS('Error in ExtDateField widget!',e);
        }
       });
    </script>
  </xsl:template>
</xsl:stylesheet>
```

/Listing (Abschnitt 2 von 2)

4.7 Mitarbeiter mit Tasks in Workflows einbeziehen

Dieser Abschnitt erläutert die folgenden Themen:

- *Tasks in Technical Workflows integrieren, S. 103*
- *Tasklisten anzeigen, S. 105*

Tasktypen

Die inubit Suite 6 bietet folgende Typen von Tasks an:

- **Formulare**

Formular-Tasks sind sinnvoll, wenn Daten von Mitarbeitern eingegeben oder kontrolliert werden sollen. Die Formulare erstellen Sie per Drag'n'Drop im Formular Designer des Task Generators.

→ Siehe *Formulare im Formular Designer erstellen (Workbench/ Process Engine: Modul-Guide, Kap. 4.2, S. 63)*.

- **Information**

  Geeignet, um Mitarbeitern kurze Mitteilungen anzuzeigen.

- **Externes Programm**

  Bei diesem Typ wird ein externes Programm geöffnet, in dem die Aufgabe erledigt werden soll.

  Beispiel: Ein File Input Connector holt eine MS Word-Datei aus dem Dateisystem, leitet diese an den Task Generator weiter und löst damit die Task „Datei in Word anzeigen" aus. Das Programm wird geöffnet, die Datei angezeigt und der Mitarbeiter bearbeitet diese. Mit dem Beenden von Word ist die Task erledigt.

  → Siehe *Dialog „Task Generator Eigenschaften" (Workbench/ Process Engine: Modul-Guide, Kap. 4.10.1, S. 124)*.

Taskzustellung

Es gibt folgende Möglichkeiten, Tasks zu Mitarbeitern zuzuordnen

- **Portalrollen**

  Wenn die Tasks den Mitarbeitern im inubit Enterprise Portal angezeigt werden sollen, dann können Sie Portalrollen nutzen.

  → Siehe *Portlet für ausgewählte Rollen sichtbar machen (Process Engine: Administrator- und Entwickler-Guide, Kap. 14.7.2, S. 177)*..

- **Prozessrollen**

  Um Mitarbeitern Tasks in einem Browser oder in der inubit Workbench anzubieten, verwenden Sie Prozessrollen.

  → Siehe *Prozessrollen verwenden (Process Engine: Administrator- und Entwickler-Guide, Kap. 6.9, S. 79)*.

- **Regelbasierte Task-Zustellung**

  Zusätzlich können Sie Business Rules vom Typ „Task-Regel" nutzen, um Tasks zu Mitarbeitern zuzuordnen.

  → Siehe

  - *Beispiel (Workbench: Benutzer-Guide, Kap. 15, S. 397)*. Sie finden dort ein Beispiel dafür, wie Business Rules verwendet werden können.
  - *Regeltyp „Task-Regel" (Workbench: Benutzer-Guide, Kap. 15.1, S. 399)*
  - *Task-, Daten oder Entscheidungs-Regel anlegen (Workbench: Benutzer-Guide, Kap. 15.4, S. 402)*

Die Portal- und Prozessrollen und Task-Regeln ordnen Sie im *Dialog „Berechtigungen" (Workbench/Process Engine: Modul-Guide, Kap. 4.10.2, S. 127)* des Task Generators zu Tasks zu.

Die Rollenzuordnungen können Sie auch zur Ausführungszeit des Task Generators dynamisch überschreiben lassen.

→ Siehe *Listen-Moduleigenschaften überschreiben (Workbench: Benutzer-Guide, Kap. 14.8.3, S. 382).*

Task-Delegation

Mitarbeiter können ihre Tasks an andere Mitarbeiter weitergeben. Sie können, genau wie bei der Taskzustellung, festlegen, für welche Tasks eine Delegation möglich sein soll, an wen die Tasks delegiert werden können und wer delegieren darf.

Um die Empfänger einer delegierten Tasks zu definieren, wählen Sie eine Business Rule vom Typ „Task-Regel“ im *Dialog „Berechtigungen“ (Workbench/Process Engine: Modul-Guide, Kap. 4.10.2, S. 127)* des Task Generators. In demselben Dialog legen Sie auch fest, welche Rollen berechtigt sind, Tasks zu delegieren.

Start und Ablauf von Tasks

Sie können für jede Task Bearbeitungszeiten festlegen. Damit bestimmen Sie, wann und wie lange die Tasks in der Tasklist angezeigt werden sollen. Der Technical Workflow, in dem die Task entstanden ist, kann warten, bis die Task erledigt ist, oder direkt nach Erzeugen der Task fortgesetzt werden.

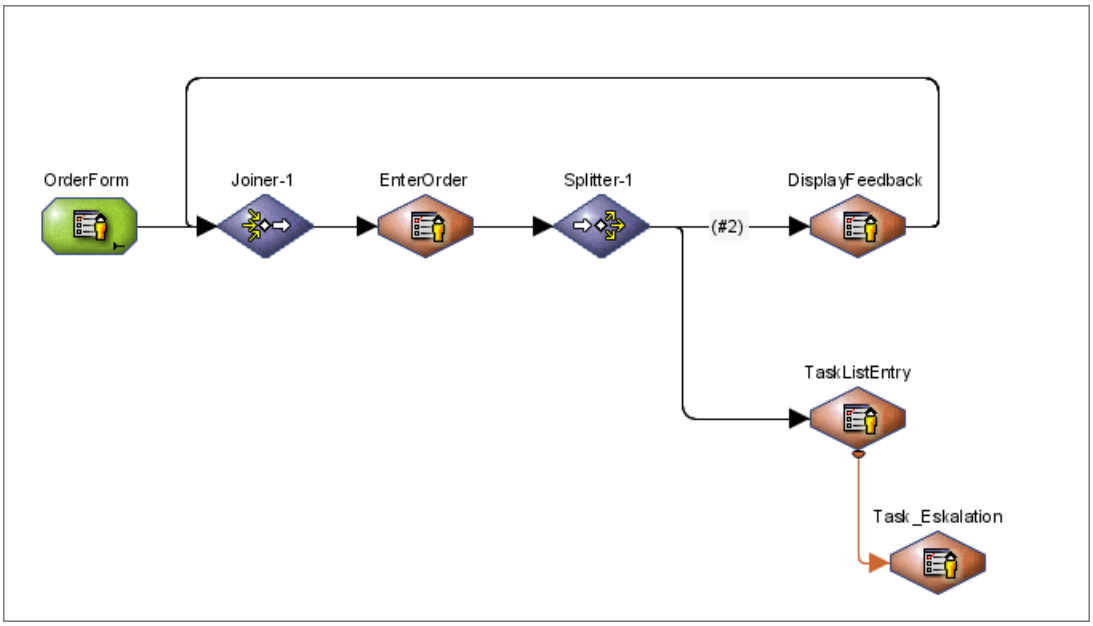
→ Diese Eigenschaften konfigurieren Sie beim Anlegen des Task Generators, siehe *Dialog „Task Generator Eigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 4.10.1, S. 124)*.

Task-Eskalation

Tasks, deren Bearbeitungszeit abgelaufen ist, können automatisch eskaliert werden.

Dazu wird das Fehlerhandling der inubit Suite 6 genutzt: der Task Generator wird so konfiguriert, dass dieser einen Fehler wirft, wenn die Bearbeitungszeit einer Task abgelaufen ist. Diesen Fehler fangen Sie durch ein weiteres Modul ab, das als Fehlerausgang mit dem Task Generator verbunden ist.

Als Fehlerausgang können Sie z. B. einen zweiten Task Generator mit dem ersten Task Generator verbinden. Der zweite Task Generator erzeugt zur Eskalation eine neue Task, sobald der Fehler geworfen wird:



Damit der erste Task Generator nach Ablauf der Bearbeitungszeit einen Fehler wirft, dürfen folgende Optionen des Task Generators **nicht** markiert sein: „Zeitüberschreitung nicht als Fehler werten“ und „Workflow trotz durchlaufenem Fehlerausgang oder -scope als erfolgreich werten“.

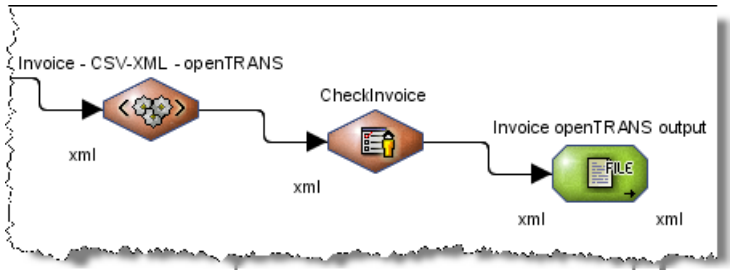
Wenn die Task, deren Bearbeitungszeit abgelaufen ist, vorher bereits delegiert worden ist, dann ist es sinnvoll, vor dem Erzeugen der neuen Task zu ermitteln, an wen die Task delegiert wurde. Dazu werten Sie die Systemvariable `ISTaskDelegatedTo` aus. Diese Variable enthält die Rolle und den Namen des Benutzers, an welchen die Task zuletzt delegiert wurde. Die Liste ist leer, wenn die Task nicht delegiert wurde.

4.7.1 Tasks in Technical Workflows integrieren

Mit Hilfe von Tasks können Sie Mitarbeiter, Kunden und Geschäftspartner in Ihre Geschäftsprozesse einbeziehen. Dazu binden Sie für jeden Task, der im Laufe des Prozesses nötig ist, einen Task Generator in den Technical Workflow ein, der Ihren Geschäftsprozess technisch realisiert.

Beispiel

In dem abgebildeten Beispiel-Workflow werden Rechnungen erzeugt, von Mitarbeitern des Controllings kontrolliert und freigegeben:



Die Rechnungsdaten werden im XSLT Converter aufbereitet und an den Task Generator übergeben. Der Task Generator erzeugt für Mitarbeiter mit der Rolle „Controlling“ eine Task vom Typ „Formular“. Die Task wird in der Taskliste angezeigt:

Mit einem Klick auf die Task übernimmt der Mitarbeiter diesen. Das Formular mit den Rechnungsdaten wird angezeigt und die Rechnung kann geprüft werden.

Wenn Sie Tasks zu Mitarbeitern über Portal- oder Prozessrollen zuordnen, müssen diese Rollen bereits vorhanden sein. Siehe *Portalrollen verwalten (Process Engine: Administrator- und Entwickler-Guide, Kap. 14.6, S. 172)* und *Prozessrollen verwenden (Process Engine: Administrator- und Entwickler-Guide, Kap. 6.9, S. 79)*

So gehen Sie vor

1. Erstellen Sie einen Task Generator.
2. Legen Sie im *Dialog „Task Generator Eigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 4.10.1, S. 124)* den Typ der Task fest.
3. Legen Sie im *Dialog „Berechtigungen“ (Workbench/Process Engine: Modul-Guide, Kap. 4.10.2, S. 127)* über die Zuordnung zu Prozess- oder Portalrollen fest, für welche Rollen Mitarbeiter die Task sichtbar sein soll.

   Lassen Sie sich von dem Modul-Assistenten durch die weitere Erstellung des Task Generators führen.
4. **Nur bei Formular-Tasks**:

   Erstellen Sie ein Formular- und Ausgangsmapping im Register „Designer“.

Falls das Register „Designer“ zum Erstellen des Formular-Mappings nicht angezeigt wird, überprüfen Sie im Task Generator-*Dialog „XSLT Mapper Einstellungen“ (Workbench/Process Engine:*

Modul-Guide, Kap. 4.10.4, S. 129) die Option „Eingangsmapping ignorieren“. Diese Option darf nicht markiert sein!

→ Siehe auch
- *Formular-Mapping erstellen (Tutorials, Kap. 7.2.1, S. 137)*
- *Ausgangs-Mapping erstellen (Tutorials, Kap. 7.2.2, S. 141)*

4.7.2 Tasklisten anzeigen

Dieser Abschnitt erläutert die folgenden Themen:

- *Taskliste in der inubit Workbench anzeigen, S. 106*
- *Tasklist als Servlet im Browser anzeigen, S. 106*

Überblick

Tasks werden den Mitarbeitern in Listenform angezeigt. Für jede neue Task wird ein weiterer Eintrag in dieser so genannten Tasklist erstellt:

Tasklisten können im inubit Enterprise Portal, in einem Browser oder in der inubit Workbench angezeigt werden:

- **Im inubit Enterprise Portal**
  → Siehe *Portlet „Tasklist“ im inubit Process Cockpit (Workbench: Benutzer-Guide, Kap. 20.9, S. 525).*
- **In der inubit Workbench**
  → Siehe *Taskliste in der inubit Workbench anzeigen (Workbench/ Process Engine: Modul-Guide, Kap. 4.7.2.1, S. 106).*
- **Im Browser**
  → Siehe *Tasklist als Servlet im Browser anzeigen (Workbench/ Process Engine: Modul-Guide, Kap. 4.7.2.2, S. 106).*

4.7.2.1 Taskliste in der inubit Workbench anzeigen

Im Register „Tasklist“ werden alle Tasks angezeigt, die für die angemeldeten Benutzer entsprechend ihren Prozess- oder Portalrollen sichtbar sind.

Für Benutzer mit der Rolle „Tasklist User“ ist in der inubit Workbench ausschließlich das Register „Taskliste“ sichtbar!

Die Tasks sind sortiert nach Rollen.

Die verschiedenen Tasktypen sind wie folgt gekennzeichnet:

- **Formular**
  Nach Klick auf den Namen der Task wird das Formular im rechten Bereich angezeigt.
- **Information**
  Die Information wird als Tooltipp der Task. Nach Klick auf den Namen der Task wird die Information aus der Liste entfernt.
- **Externe Applikation**
  Nach Klick auf den Namen der Task wird die externe Applikation geöffnet. Nach dem Schließen der externen Applikation wird die Task aus der Liste entfernt.

→ Für Informationen über das Erzeugen und Bearbeiten von Tasks siehe *Tasks in Technical Workflows integrieren (Workbench/ Process Engine: Modul-Guide, Kap. 4.7.1, S. 103)*.

4.7.2.2 Tasklist als Servlet im Browser anzeigen

Die Tasklist kann als Servlet ohne manuelle Anpassungen in folgenden Browsern angezeigt werden:

- **Windows**: Firefox ab Version 2
- **Linux**: Firefox ab Version 2

Voraussetzungen

- Benutzer müssen die Rolle „Tasklist User“ oder eine andere Rolle mit dem Recht „Tasklist“ haben.
- Im Browser müssen Cookies erlaubt sind.

Anmelden

So gehen Sie vor

1. Öffnen Sie einen Browser und geben Sie folgende URL ein:
   `http://[NamePortalserverRechner]:8000/ibis/servlet/FormServlet`
   Der Anmeldedialog wird angezeigt.

2. Geben Sie Benutzername und Passwort ein.
3. Klicken Sie auf „Anmelden“.

Die Tasklist wird angezeigt.

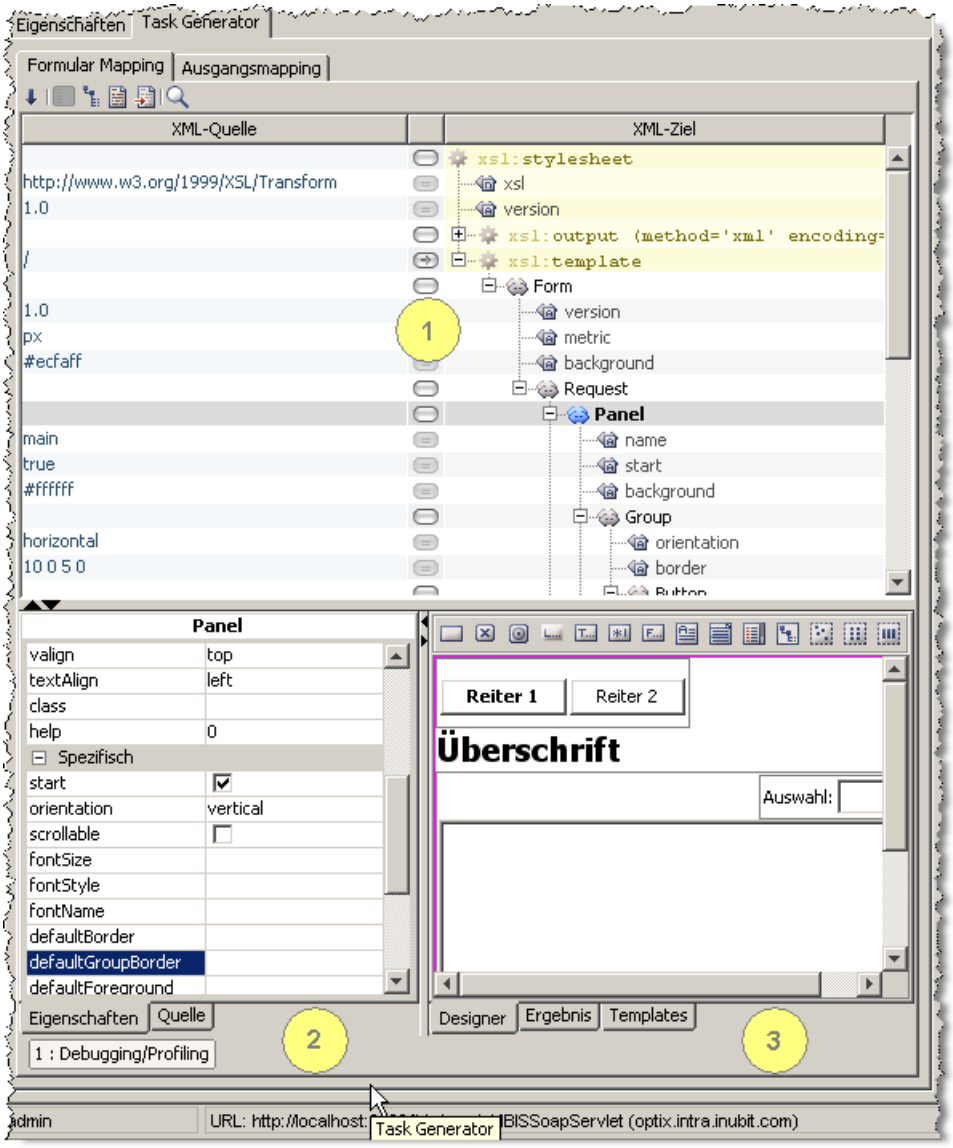
Abmelden Zum Abmelden genügt es, den Browser zu schließen.

4.8 Benutzeroberfläche des Formular Designers

Dieser Abschnitt erläutert die folgenden Themen:

- *Formular Mapping, S. 108*
- *Ausgangsmapping, S. 110*
- *Formular-Elemente in der Werkzeugleiste, S. 111*
- *Eigenschaften von Formular-Elementen, S. 114*

4.8.1 Formular Mapping



Im Register „Formular Mapping“ legen Sie fest, wie eine XML-Eingangsnachricht auf den XML-Request abgebildet wird, aus dem das Formular zur Anzeige in der inubit Workbench oder Browser erzeugt wird.

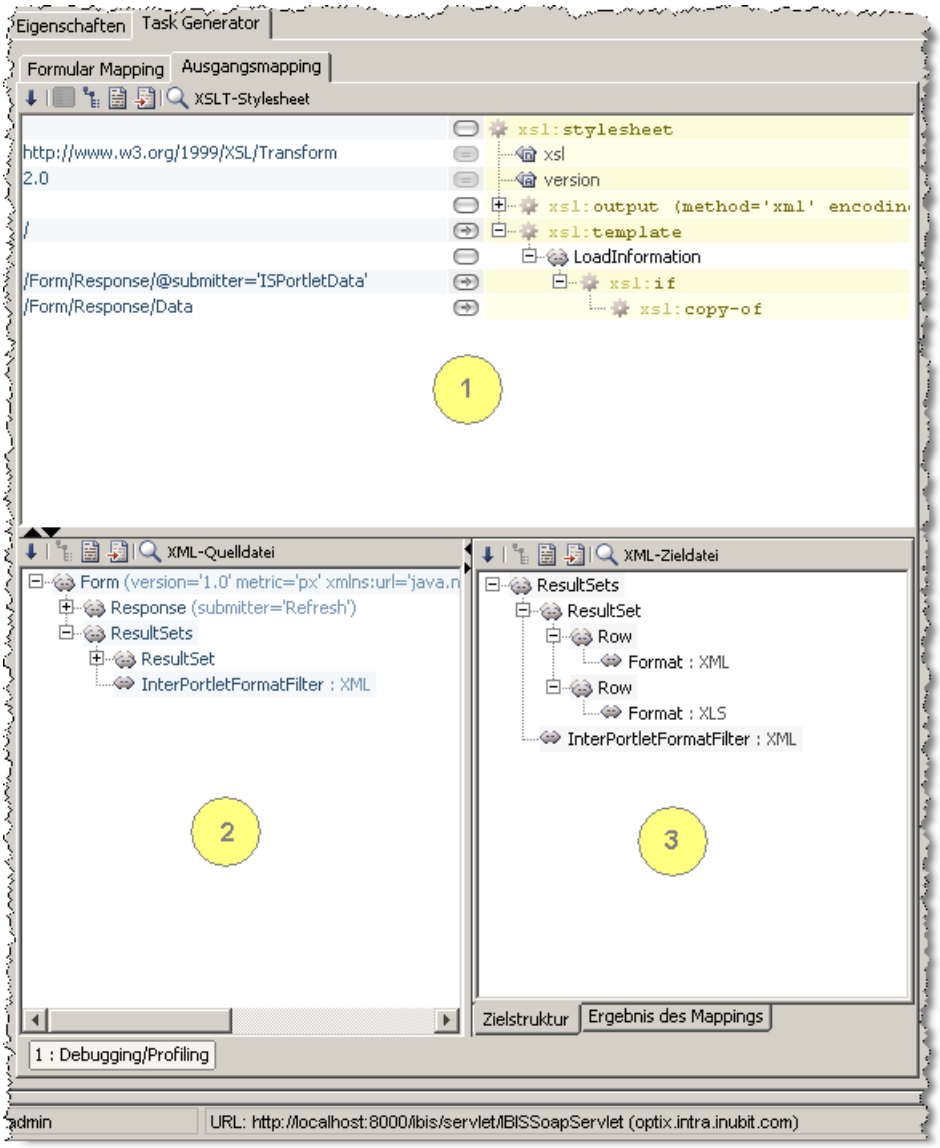
→ Siehe *Formular-Mapping erstellen (Tutorials, Kap. 7.2.1, S. 137).*

1. **Abbildungstabelle**
   Zeigt das XSLT-Stylesheet an, das Sie zum Abbilden der XML-Eingangsnachricht auf die XML-Response erstellen:
   - **XML-Quelle**
     Enthält Werte, XPath-Ausdrücke und Funktionen. Diese stammen meist aus dem Bereich „XML-Quelle“ und werden per Drag’n’Drop in die Spalte gezogen.
   - **XML-Ziel**

Enthält XSLT-Anweisungen (Elemente und Attribute), die das Ergebnis der Drag'n'Drop-Aktionen im Designer sind.

2. **Eingabe**
   - **Eigenschaften**
     Zum Festlegen der Eigenschaften des Formular-Elements, das im Register „Designer" oder im Bereich „XML-Ziel" ausgewählt ist.
     Eingaben werden mit der Eingabetaste abgeschlossen. Standardwerte stellen Sie über das Kontextmenü wieder her.
     → Siehe *Eigenschaften von Formular-Elementen (Workbench/Process Engine: Modul-Guide, Kap. 4.8.4, S. 114).*
   - **Quelle**
     Zum Laden und Anzeigen der Eingangsnachricht.
   - **Debugging/Profiling**
     → Siehe *XSLT-Stylesheets debuggen (Workbench/Process Engine: Modul-Guide, Kap. 5.3, S. 142).*
3. **Ausgabe**
   - **Designer**
     Zum Erstellen des Formulars per Drag'n'Drop aus der Werkzeugleiste in den Designer.
     → Siehe *Formular-Elemente in der Werkzeugleiste (Workbench/Process Engine: Modul-Guide, Kap. 4.8.3, S. 111).*
   - **Ergebnis**
     Zeigt das Ergebnis nach dem Testen des Mappings an.
   - **Templates**
     Enthält alle Formular-Elemente als XML.

4.8.2 Ausgangsmapping



Im Register „Ausgangsmapping“ legen Sie fest, wie die XML-Response aus dem geänderten Formular auf das XML-Format abgebildet werden, welches das nächste Modul im Workflow erwartet.

→ Siehe *Ausgangs-Mapping erstellen (Tutorials, Kap. 7.2.2, S. 141).*

1. **Abbildungstabelle**
   Zeigt das XSLT-Stylesheet an, das Sie zum Abbilden der XML-Response auf das Ausgangs-XML-Format erstellen.
2. **XML-Quelldatei**
   Zeigt den Original-Request und die Response an, die der Browser nach dem Bearbeiten und Absenden des Formulars wieder an den Task Generator gesendet hat.
3. **XML-Zieldatei**
   - **Zielstruktur**

Zum Laden einer XML-Datei, die bereits so strukturiert ist, wie vom nachfolgenden Modul erwartet.

- **Ergebnis des Mappings**
  Struktur und Inhalt der Ausgangsnachricht des Task Generators.
- **Debugging/Profiling**
  → Siehe *XSLT-Stylesheets debuggen (Workbench/Process Engine: Modul-Guide, Kap. 5.3, S. 142).*

4.8.3 Formular-Elemente in der Werkzeugleiste

Die Werkzeugleiste im Designer enthält folgende Buttons zum Erstellen von Formularelementen:

Die Erläuterungen in der folgenden Tabelle beziehen sich auf die Buttons in der Werkzeugleiste von links nach rechts:

Button	Bezeichnung	Verwendung
	Button	Zum Auslösen einer Aktion
	Checkbox	Zum Aktivieren/Deaktivieren genau einer Option in einem Gruppen-Element
	Radiobutton	Zum Auswählen von einer Option aus mehreren; es werden immer alle Optionen in einem Gruppen-Element betrachtet
	Label	Beschriftung, z. B. eines Textfields
	Textfield	Texteingabefeld
	Passwordfield	Passwort-Eingabefeld; alle eingegebenen Zeichen werden aus Sicherheitsgründen als Asterisk (*) dargestellt.
	FileField	Erzeugt Formularfelder zum Upload einer Datei (Eingabefeld und Button mit dem Label „Durchsuchen“). Beim Submit des Formulars werden die hochgeladenen Dateien mit base64-kodiert und in die XML-Response geschrieben.
	Textarea	Eingabefeld für längere Texte.
	Combobox	Aufklappbare Liste zur Auswahl von Optionen, eine der Optionen kann standardmäßig vorausgewählt sein und daher immer angezeigt werden. Die Liste ist optional auch bearbeitbar.

Button	Bezeichnung	Verwendung
	List	Liste, aus der eine Option gewählt werden kann.
	Tree	Baumdarstellung von Elementen, die geöffnet und geschlossen werden können.
	Group - Absolute Layout	Ordnet alle Formular-Elemente in dieser Gruppe pixelgenau so an, wie vom Benutzer angeordnet. Die Größe der absoluten Gruppe passt sich nicht automatisch ihrem Inhalt an; daher sollte ggf. der Scrollbar aktiviert werden (Attribut `scrollable`). Gruppen können geschachtelt werden
	Group - Table Layout	Erzeugt eine Tabelle mit initial drei Spalten und drei Zeilen. Gut geeignet, um darin Formularelemente auszurichten. Gruppen können geschachtelt werden
	Group - Horizontal Layout	Ordnet alle Formular-Elemente, die in diese Gruppe gezogen werden, horizontal untereinander an; Gruppen können geschachtelt werden
	Group - Vertical Layout	Ordnet alle Formular-Elemente, die in diese Gruppe gezogen werden, vertikal untereinander an; Gruppen können geschachtelt werden
	Filler	Erzeugt eine quadratische Fläche, deren Kantenlänge angegeben werden kann
	Filler h	Erzeugt einen horizontalen Abstand, dessen Breite angegeben werden kann
	Filler v	Erzeugt einen vertikalen Abstand, dessen Höhe angegeben werden kann
	Separator	Horizontale oder vertikale Trennlinie, abhängig vom Wert des Attributs `orientation`

Button	Bezeichnung	Verwendung
	Image	Zum Einbinden einer Grafik in das Formular. Sie haben folgende Optionen: ■ Legen Sie die Grafik im Dateisystem ab und geben Sie den absoluten Pfad als Wert des Attributs `url` an. ■ Speichern Sie die Grafik im Repository der inubit Suite 6 und binden Sie diese ein. Dabei werden die Graphiken über einen internen Proxy geroutet. Diese Option hat folgende Vorteile: - Die Grafiken werden auch im Designer angezeigt - Die Graphiken werden auch dann noch geladen, wenn die inubit Suite 6 auf einen anderen Rechner als der Portalserver installiert ist. → Siehe *Repositorydateien über URL ansprechen (Workbench: Benutzer-Guide, Kap. 19.10, S. 499).* Sie können die Größe des Image bei `width` und `height` in Prozent angeben (z. B. 50%) und die `border` in der angegeben Weise mit 4 Werten für oben, unten, rechts und links.

	Web	(nur bei Theme „Web“)
		Fügt ein `Web`-Element in das XSLT-Stylesheet ein. Geben Sie als Inhalt des Elements beliebiges XHTML ein. Das XHTML wird direkt in die HTML-Seite kopiert, z. B.
		`<Web>`
		`<div align="center">`
		`<object type="application/x-shockwave-flash" data="inubit.swf" width="700" height="400" title="inubit AG">`
		`<param name="movie" value="inubit.swf">`
		`<param name="quality" value="high">`
		`</object></td>`
		`</div>`
		`</Web>`

	Panel	Fügt ein neues Panel ein. → Siehe *Formular-Layout festlegen und Elemente auf Panels positionieren (Workbench/Process Engine: Modul-Guide, Kap. 4.2.1, S. 63).*
	Theme	Legt fest, wie das Formular dargestellt wird: ■ „Toolset“ geeignet für Formulare, die in der Tasklist der inubit Workbench angezeigt werden. ■ „Web“ ist geeignet für browser-basierte Formulare.

4.8.4 Eigenschaften von Formular-Elementen

Im Register „Eigenschaften“ werden die Eigenschaften des aktuell im Formular markierten Elements angezeigt. Die Eigenschaften sind in folgende Bereiche gruppiert:

- *Bereich „Allgemein“ (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 114)*
- *Bereich „Spezifisch“ (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 115)*
- *Bereich „Abhängig“ (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 119)*

Bereich „Allgemein“

- **`name`**

  Zum Referenzieren des Elements über dessen Namen. Der Name darf nur XML-konforme Zeichen enthalten, weil der Name als XML-Tag in der Response verwendet wird.

  Solange kein Name vergeben ist, wird automatisch eine ID vergeben, z. B. „CheckBox_7592354“.

  → Sie können Elemente alternativ über deren Position referenzieren, siehe `path` *(Workbench/Process Engine: Modul-Guide, Kap. 4, S. 114)*.

Sie können das `name`-Attribut nur dann nutzen, wenn das Element kein `path`-Attribut hat.

- **`path`**

  Zum Referenzieren des Elements über dessen Position.

  Mit dem `path`-Attribut können Sie z. B. mehrdimensionale Strukturen aufbauen, da die Felder in der Response in der entsprechenden XML Struktur erzeugt werden.

  Beispiel:

  - `path=/Bestellung|0/Waren|0/Buch|1.`
  - In der Response wird folgende Struktur ausgegeben:

```
<Path>
<Bestellung>
 <Waren>
   <Buch>false</Buch>
 </Waren>
</Bestellung>
</Path>
```

Der Index bezieht sich nicht auf das Element diesen Namens wie bei XPath, sondern auf alle Elemente auf dieser Ebene. Damit wird die komplette Reihenfolge wiederhergestellt.

→ Sie können Elemente alternativ über deren Namen referenzieren, siehe `name` *(Workbench/Process Engine: Modul-Guide, Kap. 4, S. 114)*.

Sie können das `path`-Attribut nur dann nutzen, wenn das Element kein `name`-Attribut hat

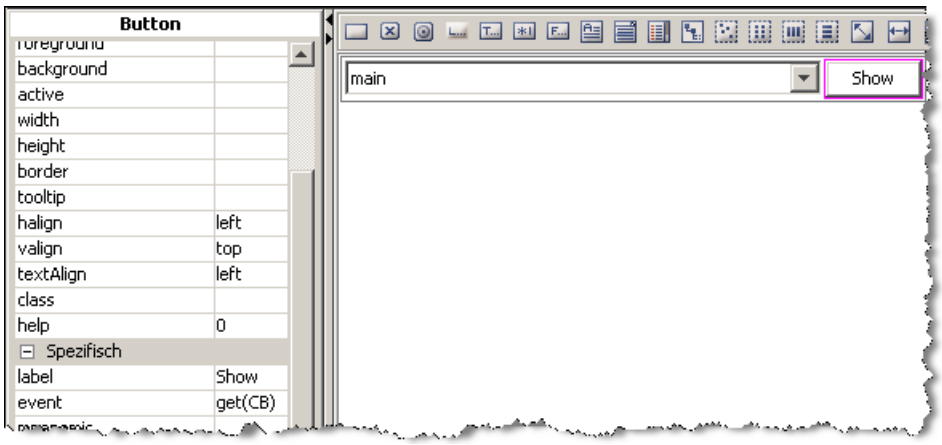
- **`enabled`**
  Zum Aktivieren/Deaktivieren des Formularelements.
- **`foreground`**
  Wenn Sie in dieses Feld klicken, erscheint am Ende der Eingabezeile ein Button. Ein Klick auf diesen Button öffnet einen Farbwähler. Mit dieser Farbe bestimmen Sie die Textfarbe.
- **`background`**
  Hintergrundfarbe des Elements.
- **`width`**
  Breite des Elements (ganze Zahl). Bei einer unzulässigen Eingabe wird die Breite auf „1“ gesetzt.
- **`height`**
  Höhe des Elements (ganze Zahl).
- **`border`**
  Breite des Rahmens, der um das Element gezeichnet werden soll (ganze Zahl). Die Farbe des Rahmens entspricht der Farbe, die als Hintergrundfarbe des Panels angegeben wurde.
- **`tooltip`**
  Text, der angezeigt wird, wenn der Mauszeiger im Formular auf das Element zeigt. Ein Tooltipp besteht aus nicht mehr als 80 Zeichen, da einige Web-Browser längere Texte nicht anzeigen können.
- **`halign`**
  Waagerechte Ausrichtung: linksbündig (left), zentriert (center) oder rechtsbündig (right).
- **`valign`**
  Senkrechte Ausrichtung, oben (top), zentriert (center) oder unten (bottom).

Bereich „Spezifisch“

Wenn ein Element „Panel“ markiert ist, werden über die spezifischen Attribute die Eigenschaften aller Elemente beeinflusst, für die keine Werte angegeben sind.

Darüber hinaus gibt es noch eine Reihe weiterer „spezifischer“ Attribute. Die folgende Aufzählung listet die Attribute und Werte, die nicht selbsterklärend sind:

- **label**
  Beschriftung eines Elements, z. B. eines Buttons.
- **event**
  Wählen Sie ein Event aus oder geben Sie den Namen des Panels ein, das angezeigt werden soll, wenn das Event ausgelöst wird.
  Es gibt folgende Events:
  - **<Panel_Name>**
    Zum Navigieren von einem Panel zum nächsten, wird verwendet z. B. an einem „Weiter“-Button.
    → Siehe *Formular-Layout festlegen und Elemente auf Panels positionieren (Workbench/Process Engine: Modul-Guide, Kap. 4.2.1, S. 63).*
  - **back()**
    Sprung zurück zum zuletzt angezeigten Panel.
  - **submit(value)**:
    Zum Absenden des Formulars. Die im Formular enthaltenen Daten werden als XML-Nachricht an den Task Generator übergeben.
    Wenn in einem Formular mehrere Buttons vorhanden sind, dann ersetzen Sie „value“ durch eine beliebige Zeichenkette, damit Sie den Auslöser des Events identifizieren und Folgeaktionen bestimmen können. Die Zeichenkette wird als Wert des Attributs `submitter` in der Response ausgegeben.
  - **cancel()**
    Bricht die Anzeige des Formulars ab. Die Task wird weiterhin in der Taskliste angezeigt.
  - **get(componentName)**
    Sucht das Element, dessen Name in Klammern angegeben ist und wertet den aktuell angezeigten Inhalt des Elements aus.
    Beispiel:
    Die Combobox `CB` enthält eine Liste mit den Namen aller vorhandenen Panels. Für den Button „Show“ ist das Event `get(CB)` definiert. Wenn Benutzer einen Listeneintrag auswählen und dann auf den Button klicken, dann wird der aktuell angezeigte Wert der Combobox ausgewertet und das dazugehörige Panel angezeigt.



- **`browse()`**
  Öffnet einen Datei-Explorer. Es wird der Pfad und Dateiname der ausgewählten Datei zurückgeliefert und in ein Textfeld geschrieben.
  Nur für Formulare, die in der inubit Workbench angezeigt werden!
- **`execute(profileName|URL)`**
  Bei Formularen, die von der inubit Workbench angezeigt werden, muss der Name eines Profils angegeben werden, das ausgeführt werden soll. Der Profilname muss dem Namen eines der Applikationsprofile entsprechen, die im Register „Configuration > Applikationsprofile“ definiert sind.
  Bei Formularen, die im Web-Browser angezeigt werden, muss eine URL angegeben werden.

- **`validate`**
  → Siehe *Server-seitige Validierung von Benutzereingaben per AJAX (Workbench/Process Engine: Modul-Guide, Kap. 4.3.2, S. 82)*.
- **`stateChange`**
  Ändert nach Absenden des Formulars die Größe des Portlets entsprechend dem gewählten Wert. Diese Funktion kann z. B. genutzt werden, um ein Portlet minimiert und damit platzsparend anzubieten und es erst nach einer Benutzereingabe auf die vollständige Größe zu skalieren.
- **`ajax`**
  Aktualisiert den Inhalt des Formulars, in dem sich der Button befindet, ohne die komplette Webseite neu zu laden.
  Diese Funktion beschleunigt die Reaktion von Webanwendungen auf Benutzeraktionen und verhindert, dass statische Daten erneut übertragen werden müssen, die sich nicht geändert haben.

Verwenden Sie die Optionen **`ajax`** und **`ajaxGroup`** nicht im Rahmen einer Interportlet-Kommunikation und nicht in Gruppen,

die ein „FileField“ zum Hochladen einer Datei enthalten! Die Optionen sind in diesen Kontexten nicht funktional.

Mit dem Ajax-Submit können Sie Formulare auch teilweise statt vollständig aktualisieren lassen, siehe *Portlets und Formularabschnitte mit Ajax aktualisieren (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 76)*.

- **`ajaxGroup`**
  Um den Inhalt der ausgewählten Formular-Gruppe zu aktualisieren, ohne die komplette Webseite neu zu laden.
  → Siehe *Portlets und Formularabschnitte mit Ajax aktualisieren (Workbench/Process Engine: Modul-Guide, Kap. 4.2.10, S. 76).*
- **`jump`**
  Zum Festlegen der Einsprungstelle in dem Technical Workflow, ab welcher die Daten für die Ajax-Gruppen-Aktualisierung aufbereitet werden.
  → Siehe *Formularabschnitt mit Ajax aktualisieren (Workbench/ Process Engine: Modul-Guide, Kap. 4.2.10.2, S. 78).*
- **`mnenomic`**
  (Nur für Formulare, die in der Tasklist der inubit Workbench angezeigt werden)
  Elemente, die Ereignisse auslösen können, wie z. B Buttons, werden automatisch mit einem Shortcut belegt. Dabei wird der Buchstabe des Element-Labels unterstrichen, mit dem das Ereignis ohne Mausklick, nur über die Tastatur ausgelöst werden kann. Bei Linux wird der unterstrichene Buchstabe immer angezeigt. Bei Windows müssen Sie erst die Alt-Taste drücken, damit die Shortcuts angezeigt werden.
- **`Panel scroll „true, false“`**
  Standardmäßig liegt das Panel in einem Container, der automatisch mit horizontalem und vertikalem Scrollbalken ausgestattet ist. So kann immer das ganze Formular bearbeitet werden, auch wenn der Bildschirm kleiner als das Panel klein ist.
  Mit `scroll="false"` werden die Scrollbalken abgeschaltet.
- **`mandatory`**
  Kennzeichnet das Element als Pflichtfeld.
- **`pattern`**
  Enthält einen regulären Ausdruck, der das Muster der erlaubten Benutzereingaben definiert. So können Sie z. B. die Struktur einer Nummer mit zwischengeschalteten Sonderzeichen und einer bestimmten Anzahl von Zahlen definieren.
  Als Anfangs- und Endzeichen des regulären Ausdrucks müssen Sie Schrägstriche setzen, z. B. `/(\d\d).(\d\d).(\d\d\d\d)/`.

Das Attribut `pattern` kann nicht zusammen mit dem Attribut `mandatory` verwendet werden! Wenn die Benutzereingabe in einem Feld mit `pattern` beschränkt wird und das Feld gleichzeitig ein Pflichtfeld sein soll, dann müssen Sie dies mit einem geeigneten regulären Ausdruck formulieren.

- **`TextArea searchText='Suchstring'`**
  (Nur für die Taskanzeige in der inubit Workbench)
  Sucht und markiert den Text in der Komponente
- **`TextArea Highlight`**
  (Nur für die Taskanzeige in der inubit Workbench)
  Unter TextArea kann beliebig oft das Element Highlight stehen. Dies hat folgende Attribute:
  - start='Ganzzahl'
  - stop=' Ganzzahl'

  Sie können entweder `stop` oder `length` benutzen!
  - length=' Zahl'
  - color='#xxxxxx'
- **`Tree searchXPath`**='Expression':
  (Nur für die Taskanzeige in der inubit Workbench)
  Sucht und markiert den entsprechenden XPath, z. B. `/*/Row[3]`
- **`disableBack`**
  Attribut des Form-Elements. Wenn die Option markiert ist, dann wird in einer Formularfolge die Nutzung des „Back“-Buttons verhindert. Nach einem Klick auf den „Back“-Button versucht der Browser, sofort wieder das Formular anzuzeigen, das den angeklickten „Back“-Button enthält.

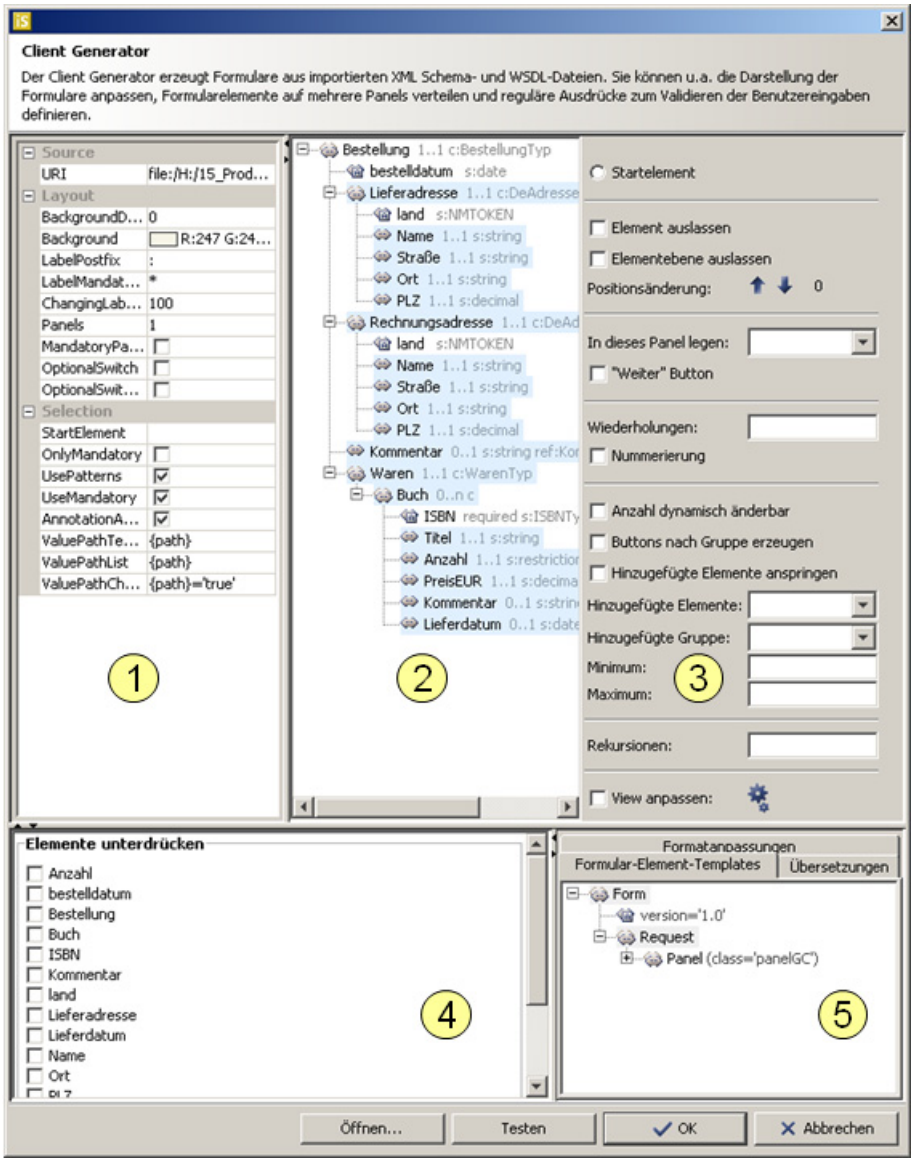
Für diese Funktion muss JavaScript aktiviert sein.

Bereich „Abhängig“

- **`visible`**
  Das Element ist nur für Portal-Benutzer sichtbar, die eine Rolle mit den angegebenen Rechten haben.
- **`enabled`**
  Das Element kann nur von Portal-Benutzern geändert oder angeklickt werden, wenn diese eine Rolle mit den angegebenen Rechten haben.

→ Siehe *Formularelemente abhängig von Rechten machen (nur bei Web-Applikationen) (Workbench/Process Engine: Modul-Guide, Kap. 4.2.7, S. 71).*

4.9 Benutzeroberfläche des Client Generators



Der Client Generator besteht aus folgenden Bereichen:

1. **Basis-Einstellungen**
   Alle Einstellungen in diesem Bereich sind dateiweit gültig.
   - **Source**
     Pfad zu der geladenen Schema- oder WSDL-Datei.
   - **Layout**
     - **`BackgroundDelta/Background`**
       `Background` legt die Farbe des Hintergrunds bei verschachtelten group-Elementen fest; wird aktiviert, sobald `BackgroundDelta` definiert ist und mind. zwei ineinander verschachtelte group-Elemente vorhanden sind. Bei

positiven Werten für `BackgroundDelta` lässt die Helligkeit der Hintergrundfarbe bei der inneren Gruppe nach, bei negativen Werten steigt sie.

- **`LabelPostfix`**
  Standardmäßig angezeigtes Zeichen nach einem Formular-Label.
- **`LabelMandatoryPostfix`**
  Zeichen, das Pflichtfelder als solche kennzeichnet. Wird für alle Elemente mit einer Kardinalität von 1…1 gesetzt.
- **`ChangingLabelWidth`**
  Die Angabe sorgt bei ineinander geschachtelten Gruppen dafür, dass die Element-Label einer Gruppe linksbündig ausgerichtet und komplett angezeigt werden können. Dazu wird für jede Gruppe der Wert des Border-Attributs vom angegebenen ChangingLabel-Width-Wert subtrahiert.
- **`Panel`**
  Anzahl der Panels im Formular. Ein Panel wird im Formular als Register dargestellt.
- **`OptionalSwitch`**
  Wenn markiert, werden optionale Gruppen (Elemente vom Typ „ComplexType“ und einer Kardinalität von „0…unbounded“) durch eine Checkbox und den Label-Namen ersetzt. Die Elemente der Gruppe werden erst angezeigt, wenn die Checkbox im Formular markiert wird.
- **`OptionalSwitchVisible`**
  Nur zu verwenden, wenn **`OptionalSwitch`** aktiviert ist!
  Sorgt dafür, dass beim Anzeigen des Formulars die Checkbox markiert ist und daher alle Elemente der Gruppe angezeigt werden.
- **`MandatoryParentsVisible`**
  Nur zu verwenden, wenn **`OptionalSwitch`** aktiviert ist!
  Wenn markiert, dann werden optionale Gruppen beim Anzeigen des Formulars immer angezeigt, sobald darin enthaltene Elemente Pflichtfelder sind (Kardinalität von 1…1).

- **Selection**
  - **`StartElement`**
    Zeigt das Element an, welches als Root-Element des Formulars ausgewählt wurde.
  - **`OnlyMandatory`**
    Wenn markiert, werden nur Pflichtfelder im Formular angezeigt.
  - **`UsePatterns`**: Wenn markiert, müssen die Benutzereingaben in den Formularfeldern den im Schema definierten Pattern genügen.

- **`UseMandatory`**
  Wenn markiert, werden nicht-ausgefüllte Pflichtfelder im Formular farbig hinterlegt und Submit- sowie Weiter-Buttons deaktiviert.
- **`AnnotationsAsToolTip`**
  Erzeugt Tooltipps aus im Schema vorhandenen Annotationen. Das Annotation-Element muss innerhalb des Elementes stehen, auf das es sich bezieht, und ein documentation-Element enthalten, z. B. `<complexType name="Beitrag"><annotation><documentation xml:lang="de">Alle Beiträge`

2. **Anzeige**
   Zeigt die geladene Schema- bzw. WSDL-Datei
3. **Element(gruppen)-Konfiguration**
   Zum Konfigurieren des in der Anzeige markierten Elements.
   - **Anzeigefeld**: Zeigt den XPath zum aktuell in der Anzeige markierten Element an. Alle folgenden Einstellungen werden auf dieses Element angewendet.
   - **Startelement**: Legt das markierte Element als root-Element des Formulars fest.
   - **Element auslassen**: Für das markierte Element und dessen Kind-Elemente werden keine Formular-Elemente erzeugt.
   - **Elementebene auslassen**: Für das markierte Element werden keine Formular-Elemente erzeugt. Für die Kind-Elemente dagegen werden Formular-Elemente erzeugt.
   - **Positionsänderung**: Ändert die Reihenfolge des markierten Elements und damit auch die Reihenfolge des Formularelements.
   - **In dieses Panel legen**: Legt fest, in welchem Register das markierte Element angezeigt wird. Vorher muss im Bereich 1 die Anzahl der Panels definiert worden sein.
   - **Weiter-Button:** Fügt auf dem aktuell angegebenen Panel einen Button ein, der zum nächsten Panel führt. Auf dem letzten Panel muss dieser Button nicht eingefügt werden.
   - **Wiederholungen**: Erzeugt beim Anzeigen mind. die angegebene Anzahl von Instanzen des markierten Elements. Falls die Eingangsnachricht mehr Elemente enthält, werden weitere Instanzen erzeugt. Nur auf Gruppen (ComplexTypes) anwendbar.
   - **Nummerierung**: Fügt den erzeugten Instanzen eine fortlaufende Nummerierung hinzu.
   - **Anzahl dynamisch ändern**: Fügt über dem markierten Element +/- Buttons ein, mit denen Benutzer weitere Instanzen des Elements erzeugen können. Nur auf Gruppen (ComplexTypes) anwendbar.

- **Buttons nach Gruppe erzeugen**: Fügt die Buttons nach der Gruppe statt vor dieser ein.
- **Hinzugefügte Elemente anspringen**: Scrollt das Formular an die Stelle, an welcher das Element hinzugefügt wurde. Nützlich, wenn mehr Elemente erzeugt werden können als auf eine Bildschirm-Seite passen.
- **Hinzugefügte Gruppen**: „Selektieren“ sorgt für einen roten Rahmen um das neue Element, dieser Rahmen wird erst entfernt, nachdem das Element bearbeitet wurde; „Herausstellen“ hebt das neue Element kurz farbig hervor.
- **Minimum/Maximum**: Bereits vorhandene Einschränkungen für das Vorkommen eines Elementes können manuell weiter begrenzt werden. Ein Maximum von -1 bedeutet, dass beliebig viele Elemente hinzugefügt werden können.
- **Rekursionen**: Legt bei rekursiv definierten Elementen die maximale Rekursionstiefe fest.
- **View anpassen**: Wenn markiert, kann der daneben stehende Button angeklickt werden und öffnet einen Editor, in dem die zu erzeugenden Formular-Elemente bearbeitet werden können. Z. B. kann dem Element Label ein Attribut `label` hinzugefügt werden, dessen Wert als Label im Formular verwendet wird (statt des eigentlich obligatorischen Element-Namens). Diese Änderungen bleiben auch nach dem Neu-Generieren des Formulars erhalten!

4. **Elemente unterdrücken**: Für die markierten Elemente werden (unabhängig von der Position in der Hierarchie) keine Formular-Elemente erzeugt.
5. Dieser Bereich besteht aus folgenden Registern:
   - **Formular-Element-Templates:** Enthält Vorlagen für alle zu erzeugenden Formular-Elemente. Sie können die Vorlagen und damit auch die Erzeugung der Formulare ändern.

     Über die Class-Attribute werden die CSS-Klassen festgelegt, mit denen das Layout des zu erzeugenden Formulars bestimmt wird.

Um eigene CSS-Klassen zu verwenden, referenzieren Sie in den `class`-Attributen CSS-Klassen und definieren die Werte dieser Klassen im Web Application Connector als interne Ressource.

   - **Übersetzungen**: Für Übersetzungen der Label, die aus den Elementen erzeugt werden.
   - **Formatanpassungen**: Zum Anpassen der verwendeten Datumsformate.

→ Siehe
- für Infos über die Verteilung der Elemente auf Panels: *Element(gruppen)-Konfiguration (Workbench/Process Engine: Modul-Guide, Kap. 3., S. 122)*

- *Dialog „Interne Ressourcen“ (Workbench/Process Engine: Systemkonnektor-Guide, Kap. 39.3.2, S. 403)*

4.10 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Task Generator Eigenschaften“, S. 124*
- *Dialog „Berechtigungen“, S. 127*
- *Dialog „Applikationsprofil“, S. 128*
- *Dialog „XSLT Mapper Einstellungen“, S. 129*
- *Dialog „Panel Layout“, S. 130*

4.10.1 Dialog „Task Generator Eigenschaften“

In diesem Dialog haben Sie folgende Optionen:

Direkte Formularfolge

Seite einer Web-Applikation

→ Siehe

- *Eigene Web-Applikation entwickeln (Tutorials, Kap. 7.2, S. 132)*
- *Web Application Connector (Workbench/Process Engine: Systemkonnektor-Guide, Kap. 39, S. 397)*

Eintrag in der Taskliste

- **Formular**
  → Siehe *Formulare (Workbench/Process Engine: Modul-Guide, Kap. ■, S. 100)*..
- **Externes Programm**
  → Siehe *Externes Programm (Workbench/Process Engine: Modul-Guide, Kap. ■, S. 101)*..
- **Information**
  → Siehe *Information (Workbench/Process Engine: Modul-Guide, Kap. ■, S. 101)*..

Task-Properties

Es werden nicht immer alle Einstellungen angezeigt; welche Einstellungen angezeigt werden, hängt vom ausgewählten Task-Typen ab!

- **Name der Task**
  Name, unter dem die Task angezeigt wird.
- **Beschreibung**
  Kurzer Text, der in der Taskliste zur Erläuterung der Task angezeigt wird.

Bei Tasks vom Typ „Information“ enthält das Feld „Beschreibung“ die Information, die den Mitarbeitern angezeigt werden soll.

- **Zeitpunkt der Erstellung**
  Optionen:
  - wenn das Modul ausgeführt wird,
  - zu einem festgelegten Zeitpunkt in der Zukunft.
- **Warte auf nächsten Task** (Sek.)
  Das aktuelle Formular wird maximal für die angegebene Zeitspanne angezeigt.
  Sinnvoll z. B., wenn die aktuelle Task abgeschlossen sein muss, bevor die nächste Task demselben Benutzer ohne vorherigen Eintrag in der Taskliste angezeigt wird (Option „Formular direkt anzeigen“).
- **Workflow-Ausführung anhalten**
  Wenn markiert, dann wird die Workflow-Ausführung erst fortgesetzt, nachdem die Task ausgeführt wurde.
  Wenn die Option nicht markiert ist, dann wird die Task ausgelöst und die Workflow-Ausführung sofort fortgesetzt. In diesem Fall kann die Response der Task im Workflow nicht weiterverarbeitet werden, weil zum Zeitpunkt ihres Eintreffens der Workflow bereits weiter fortgeschritten ist.

Im Test-Modus wartet der Technical Workflow immer auf die Ausführung einer Task.

- **Formular direkt anzeigen (ohne vorherigen Tasklist-Eintrag)**
  Wenn markiert, dann wird das Formular angezeigt, ohne vorher die Taskliste anzuzeigen. Diese Option ist sinnvoll, wenn das aktuell konfigurierte Modul ein Folge-Formular zu einem Formular erzeugt, welches in der Tasklist angezeigt wird.
  Benutzer-Interaktionen auf mehrere Formulare zu verteilen, ist sinnvoll, um die Interaktion mit der inubit Process Engine zu ermöglichen.
- **Task nur dem Bearbeiter der vorangegangenen Task zuweisen**

Tasks sind einer Prozessrolle und nicht einer Person zugeordnet. Mit dieser Option, wird sichergestellt, dass ein Formular nur demjenigen angezeigt wird, der das vorangegangene Formular bearbeitet hat. Dabei wird intern der Login-Name des Benutzers abgefragt, um zu verhindern, dass Benutzern mit derselben Prozessrolle dieses Formular angezeigt wird.

- **Timeout**

  Wenn die als Timeout angegebene Zeit überschritten ist, geschieht folgendes:

  - Die Task wird aus der Tasklist entfernt, vorausgesetzt, diese ist nicht gerade in Bearbeitung.
  - Die Workflow-Ausführung wird abgebrochen.
  - Der Workflow wird im Queue Manager mit einem Fehler aufgelistet, wenn die Option „Zeitüberschreitung nicht als Fehler werten“ **nicht** markiert ist,
  - Wenn ein Fehlerausgang an das Task Modul angeschlossen ist, dann wird dieser ausgeführt.
  - Wenn die Option „Zeitüberschreitung nicht als Fehler werten“ **nicht** markiert ist, dann wird eine E-Mail an die unten angegebene Adresse versendet.

  Aus dem Timeout und dem Zeitpunkt der Erstellung berechnet sich der Zeitpunkt, an dem die Tasks ablaufen. Dieser Wert wird im Feld „Ablauf“ in der Taskliste angezeigt.

- **Timeout danach, wenn gesperrt**:

  Wenn ein Task in Bearbeitung ist, dann läuft der Timeout weiter. Nach Überschreiten des Timeouts wird ein zweiter Timeout mit der angegebenen Dauer gestartet. Damit können Sie sicher stellen, dass eine Task nicht während der Bearbeitung gelöscht wird, sondern die Arbeit an der Task abgeschlossen und gespeichert werden kann.

- **Bei Timeout E-Mail senden an**

  E-Mailadresse, an die eine E-Mail gesendet wird, wenn die Task nicht bis zu der im Timeout angegebenen Zeit erledigt ist.

Damit die E-Mail gesendet werden kann, müssen Sie die E-Mail-Benachrichtigung aktivieren. Siehe *E-Mail-Benachrichtigung aktivieren (Process Engine: Administrator- und Entwickler-Guide, Kap. 9.2, S. 121).*

- **Zeitüberschreitung nicht als Fehler werten**

  Wenn markiert, dann wird nach Überschreiten des angegebenen Timeout keine E-Mail verschickt und kein Eintrag im Queue Manager erstellt. Die Task wird aus der Taskliste entfernt.

4.10.2 Dialog „Berechtigungen“

In diesem Dialog legen Sie fest, wer die Tasks erhalten soll.

Taskzustellung

Nur für die Tasktypen „Formular“, „Information“ und „Externes Programm“.

- **Für jeden Benutzer einen eigenen Task erzeugen**
  - Wenn markiert, dann wird für alle Benutzer eine eigene Task erstellt, die eine der ausgewählten Portal- oder Prozessrollen haben. Wenn es 10 Benutzer mit der Rolle X gibt, dann erhalten demnach alle zehn Benutzer eine Task.
  - Wenn nicht markiert, dann wird nur eine Task erstellt und allen Benutzern angezeigt, die eine der ausgewählten Portal- oder Prozessrollen haben. Sobald einer der Benutzer die Task übernimmt, wird die Task aus der Taskliste der anderen Benutzer entfernt.
- **Verwende Portalrollen**

  Blendet eine Liste aller vorhandenen Portalrollen ein.

  Die Task wird allen Rollen zugeordnet, deren Checkbox markiert ist.
- **Verwende Prozessrollen**

  Blendet eine Liste aller vorhandenen Prozessrollen ein.

  Die Task wird allen Rollen zugeordnet, deren Checkbox markiert ist.
- **Task-Regel**

  Um Mitarbeitern Tasks über eine Task-Regel zuzuordnen.

  → Siehe *Regeln auf technischer Ebene zuordnen (Workbench: Benutzer-Guide, Kap. 15.7, S. 406).*

Task Delegation

Nur für die Task-Typen „Formular“, „Information“ und „Externes Programm“.

- **Task-Regel**

  Um festzulegen, an wen Besitzer einer Task ihre Task delegieren dürfen. Wählen Sie eine Task-Regel aus.

  → Siehe
  - *Regeln auf technischer Ebene zuordnen (Workbench: Benutzer-Guide, Kap. 15.7, S. 406)*
  - *Task-Delegation (Workbench/Process Engine: Modul-Guide, Kap. 4, S. 102)*
- **Delegation nur ermöglichen für Rolle**

  Zum Einschränken der Erlaubnis, Tasks an andere Mitarbeiter weiterzuleiten. Nur Inhaber der ausgewählten Rolle dürfen Tasks delegieren.

Die Rolle mit dem Recht, Tasks zu delegieren, muss sinnvollerweise auch das Recht haben, die Tasks zu sehen. Achten Sie deswegen darauf, dass die Rolle auch im Bereich „Taskzustellung“ angegeben ist!

Web Application Connectoren

In diesem Bereich werden alle bereits publizierten Web Application Connectoren angezeigt.

Sie können die Eigenschaften dieser Web Application Connectoren im Formular nutzen (z. B. Rechte, CSS-Klassen und JavaScript-Code). Markieren Sie dazu die Checkbox eines Web Application Connectors.

→ Siehe

- *Formularelemente abhängig von Rechten machen (nur bei Web-Applikationen) (Workbench/Process Engine: Modul-Guide, Kap. 4.2.7, S. 71)*
- *inubit Enterprise Portal verwenden (Workbench: Benutzer-Guide, Kap. 20, S. 505)*
- *Prozessrollen verwenden (Process Engine: Administrator- und Entwickler-Guide, Kap. 6.9, S. 79)*
- für Infos über das Erzeugen der Regelsätze: *Regeln für die Task-Zustellung und Delegation anzeigen (Workbench: Benutzer-Guide, Kap. 8.3, S. 230)*

4.10.3 Dialog „Applikationsprofil“

Dieser Dialog wird für den Tasktyp „Externes Programm“ angezeigt.

Wenn mehrere Instanzen des externen Programms geöffnet sind, z. B. mehrere Fenster von Microsoft Word, dann werden Änderungen in der Datei, die als Task bearbeitet werden soll, nicht übernommen. MS Word: Wenn bereits eine Instanz von MS Word geöffnet ist und Word über die Taskliste aufgerufen wird, so hält die Workflow-Ausführung wegen der bereits geöffneten MS Word-Instanz nicht an, sondern läuft weiter. Damit werden Änderungen nicht im Workflow weitergeleitet, sondern gehen verloren!
Daher darf das Programm, dass durch die Task aufgerufen werden soll, vor diesem Aufruf noch nicht geöffnet sein!

Applikation

- **Applikationsprofil**
  Wählen Sie ein Applikationsprofil aus.
  → Siehe *Applikationsprofile (Process Engine: Administrator- und Entwickler-Guide, Kap. 11.1, S. 135).*

- **MIME-Typ**
  Die Angabe des Mime-Typs ist nötig, wenn die Task im Browser angezeigt und das externe Programm aus dem Browser heraus geöffnet werden soll.
- **Suffix**
  Dateinamenerweiterung, z. B. bei MS Word-Dateien `.doc`. Nicht alle Programme erkennen beim Zugriff auf eine Datei, ob diese durch externe Programme geöffnet werden kann.
  - Dateierkennung durch das externe Programm
    Es gibt Programme, die nur anhand des Suffix bestimmen, ob eine Datei geöffnet werden kann. Für diese Programme ist es wichtig, ein Suffix anzugeben, der dann während der Workflow-Ausführung an die Datei gehängt wird. Der Punkt vor dem Suffix muss ebenfalls angegeben werden.
  - Automatisches Anhängen der Dateiendung
    Einige Programme hängen automatisch ein Suffix an den Dateinamen, wenn keiner vorhanden ist. Wenn das korrekte Suffix bereits angegeben ist, wird es nicht nochmal angehängt. Wenn kein Suffix angegeben ist und die aufgerufene Applikation dann automatisch ein Suffix an die Datei anhängt, haben Ein- und Ausgabedatei nicht mehr denselben Namen. Dann hat die Ausgabedatei denselben Inhalt wie die Eingabedatei und die Änderungen durch den Benutzer werden nicht sichtbar. Geben Sie daher zur Sicherheit immer das Suffix an.

4.10.4 Dialog „XSLT Mapper Einstellungen“

In diesem Dialog haben Sie folgende Optionen:

Allgemein

- **Eingangsnachricht ignorieren**
  Wenn diese Option ausgewählt ist, wird der Bereich „Quelle“ in der Benutzeroberfläche des Formular Designers ausgeblendet. In diesem Fall werden ausschließlich die Formulareingaben der Benutzer als Eingabedaten verwendet.
- **Quell- und Zieldateien auf dem Server speichern**
  Normalerweise werden die Quell- und Zieldateien nur auf dem Client-Rechner gespeichert. Das Speichern auf der inubit Process Engine hat folgende Vor- bzw. Nachteile:
  - Vorteil: Das Modul kann auch auf anderen Rechnern bearbeitet werden.

- Nachteil: Die Dateien werden bei jeder Ausführung des Moduls geladen. Wenn die Dateien sehr groß sind, kann sich dies nachteilig auf die Performance auswirken.

Wenn Sie die Option nach dem Bearbeiten und Publizieren wieder abwählen, werden die Dateien von der inubit Process Engine gelöscht, genauso wie die Referenz zur lokalen Kopie auf dem Client-Rechner. Die Dateien sind dann beim nächsten Bearbeiten leer.

Formularmapping (Eingang)/ Ausgangsmapping

- **XSLT Prozessor**
  Zum Auswählen eines XSLT-Prozessors. Die XSLT-Prozessoren unterscheiden sich hinsichtlich des von ihnen unterstützten Sprachumfangs.

Statt eines mitgelieferten XSLT-Prozessors können Sie einen beliebigen anderen XSLT-Prozessor verwenden.

- **XSLT-Stylesheet cachen**
  Wenn das Modul kurz hintereinander mehrfach aufgerufen wird, dann steigert das Cachen des Stylesheets die Performanz.

→ Siehe

- *XSL-Versionen und EXSL-Unterstützung (Workbench/Process Engine: Modul-Guide, Kap. 5, S. 133)*
- *Beliebige XSLT-Prozessoren einbinden (Workbench/Process Engine: Modul-Guide, Kap. 5.6, S. 145)*

4.10.5 Dialog „Panel Layout“

Mit der Auswahl in diesem Dialog legen Sie das Basis-Layout Ihres Formulars fest. Sie können das Layout nach dem Anlegen des Moduls manuell ändern.

Layout-Vorlage

- **Vertikales Layout**
  Alle Elemente, die Sie per Drag‘n‘Drop in das Panel ziehen, werden automatisch vertikal untereinander platziert.
- **Tabellen-Layout**
  Fügt ein Tabellen-Element in das Panel ein. Sie können alle weiteren Elemente in die Tabellenzellen einfügen und dort anordnen.
- **Absolutes Layout**
  Ermöglicht es, alle Elemente per Drag‘n‘Drop exakt auf dem Panel zu platzieren.

- **Client Generator**
  Öffnet den Client Generator zum Erstellen eines XML Schema-basierten Layouts.
  Das vom Client Generator erzeugte Layout wird automatisch in ein Group-Element im Panel eingefügt.
  → Siehe *Benutzeroberfläche des Client Generators (Workbench/ Process Engine: Modul-Guide, Kap. 4.9, S. 120).*

Dieser Abschnitt erläutert die folgenden Themen:

- *Benutzeroberfläche: XSLT Mapper, S. 134*
- *XSLT-Stylesheets testen, S. 142*
- *XSLT-Stylesheets debuggen, S. 142*
- *Ergebnis validieren, S. 143*
- *Extensions und Java-Packages verwenden, S. 143*
- *Beliebige XSLT-Prozessoren einbinden, S. 145*
- *Variablen verwenden, S. 146*
- *Dialog „XSLT Converter Eigenschaften“, S. 148*

Verwendung

Mit Hilfe des XSLT Converters lassen sich XML-Dateien in beliebige Dateiformate transformieren. Zum Transformieren verwendet der XSLT Converter ein oder mehrere XSLT-Stylesheets, die angeben, wie der Inhalt der XML-Dateien transformiert wird.

Ein XSLT Converter unterstützt Sie bei folgenden Aufgaben:

- Dateiformate transformieren

  z. B. von einem XML-Format in ein anderes
- Nachrichten mit Hilfe von Templates für folgende Systemkonnektoren erstellen

  Web Services Connector, Topcall Connector, SAP Connector, JCA Connector, Datenbank Connector
- Beliebige Nachrichten anhand von Templates aus dem Repository erstellen

XSL-Versionen und EXSL-Unterstützung

Der Funktionsumfang, den Sie im XSLT Converter nutzen können, ist abhängig von dem XSL-Prozessor, den Sie verwenden. Die mitgelieferten XSLT-Prozessoren unterstützen folgende Funktionen:

XSL-Prozessor		Xalan 2.7	XSLTC	Saxon 6	Saxon 9 (Standard)
XSLT 1.0		x	x	x	(x)
XSLT 2.0					x
EXSLT	common	x	x	x	x
	sets	x	x	x	x
	math	x	x	x	x
	dates-and-times	x	x	x	x
	functions	x		x	
	random				x
	string	x	x		

XSL-Prozessor	Xalan 2.7	XSLTC	Saxon 6	Saxon 9 (Standard)
inubit XSLT Extensions (u. a. Funktionen zum Konvertieren eines Datums, zur Rückgabe von Linefeeds und anderen Steuerzeichen, zum Erzeugen von Tags und Ersetzen von Umlauten)	x	x	x	x

Wenn gewünscht, können Sie die mitgelieferten XSLT-Prozessoren durch beliebige andere XSLT-Prozessoren ersetzen.

Beachten Sie die Dokumentation Ihres XSLT-Prozessors für Informationen über dessen Extensions.

→ Siehe
- *Beliebige XSLT-Prozessoren einbinden (Workbench/Process Engine: Modul-Guide, Kap. 5.6, S. 145)*
- *Extensions und Java-Packages verwenden (Workbench/Process Engine: Modul-Guide, Kap. 5.5, S. 143)*

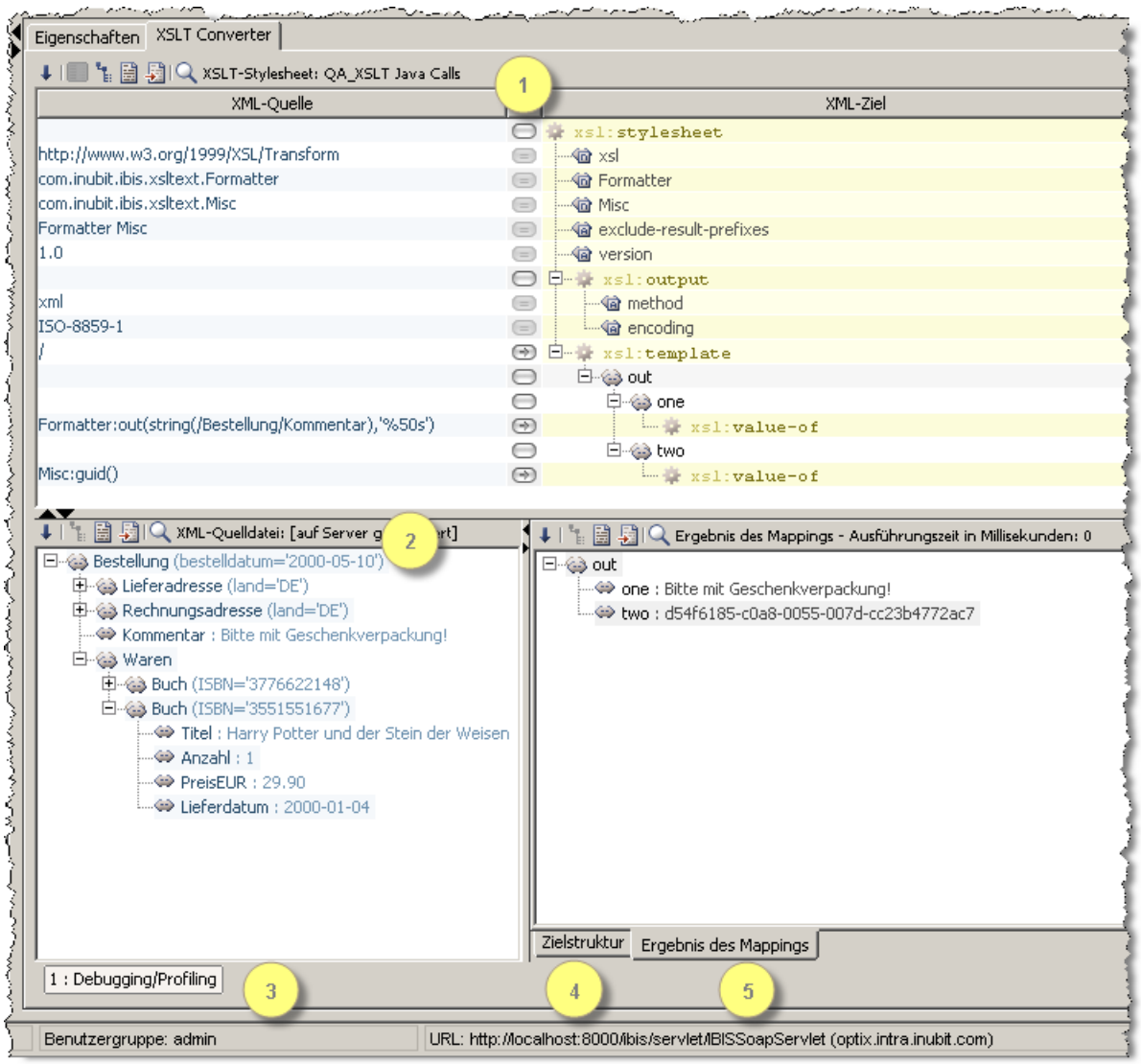
5.1 Benutzeroberfläche: XSLT Mapper

Dieser Abschnitt erläutert die folgenden Themen:

- *Funktionen in der Werkzeugleiste, S. 136*
- *Weitere Funktionen, S. 139*
- *XSLT-Kommandoassistent, S. 140*

XSLT Mapper

Der XSLT Mapper wird nach dem Erzeugen eines XSLT Converters angezeigt:



1. **XML-Quelle/XML-Ziel (Stylesheet-Bereich)**

   Zeigt das erstellte XSLT-Stylesheet als Abbildungstabelle an. Die Tabellenspalten enthalten

   - **XML-Quelle**: Werte, XPath-Ausdrücke bzw. Funktionen, meist aus dem Bereich „XML-Quelldatei“ stammend und per Drag’n’Drop in die Spalte gezogen.
   - **XML-Ziel**: XSLT-Anweisungen (Elemente und Attribute), meist aus dem Bereich „Zielstruktur“ stammend und per Drag’n’Drop in die Spalte gezogen.

   → Für Infos über die Toolbar des Bereichs siehe *Funktionen in der Werkzeugleiste (Workbench/Process Engine: Modul-Guide, Kap. 5.1.1, S. 136)*.

2. **XML-Quelldatei**

   Zum Laden und Anzeigen einer Beispiel-Eingangsnachricht, die beim Erstellen des XSLT Stylesheets als Datenquelle dient.

Der Bereich ist nicht sichtbar, wenn Sie beim Erstellen des XSLT Converters die Option „Eingangsnachricht ignorieren“ markiert haben.

→ Für Infos über die Toolbar des Bereichs siehe *Funktionen in der Werkzeugleiste (Workbench/Process Engine: Modul-Guide, Kap. 5.1.1, S. 136)*.

3. **Debugging/Profiling**
   Ein Klick auf den Button blendet den Bereich „Debugging/Profiling“ ein.
   → Siehe *XSLT-Stylesheets debuggen (Workbench/Process Engine: Modul-Guide, Kap. 5.3, S. 142)*.
4. **Zielstruktur**
   Zum Laden und Anzeigen eines Templates für die Zielstruktur. Als Template können z. B. eine beliebige Beispiel-Datei oder ein Template aus dem Repository dienen. Ein Template kann z. B. auch mit Hilfe eines SAP-, Datenbank- oder Web Service Explorers erstellt werden.
5. **Ergebnis des Mappings**
   Zeigt nach dem Testen des XSLT-Stylesheets das Ergebnis an und die benötigte Ausführungszeit.
   → Für Infos über die Toolbar des Bereichs siehe *Funktionen in der Werkzeugleiste (Workbench/Process Engine: Modul-Guide, Kap. 5.1.1, S. 136)*.

→ Siehe auch

- XSLT Converter einsetzen: *Technical Workflow erstellen 3: CSV-XML konvertieren nach openTRANS (Tutorials, Kap. 6.5, S. 116)*
- *XSLT-Stylesheets testen (Workbench/Process Engine: Modul-Guide, Kap. 5.2, S. 142)*

5.1.1 Funktionen in der Werkzeugleiste

In jedem Bereich gibt es eine Werkzeugleiste, über die Sie folgende Funktionen erreichen:

- Öffnet ein Untermenü mit folgenden Befehlen:
  - **Neu**: Löscht den aktuellen Inhalt und erstellt ein neues Dokument mit dem Element „root“
  - **Öffnen**: Zum Laden einer Datei aus dem Dateisystem, aus dem Repository oder aus der Zwischenablage. Im Editor wird automatisch die zum Dateityp passende Ansicht geladen. Weiterhin zum Laden von Business Object-Diagrammen oder URLs oder von Templates aus folgenden Explorern:
    - Datenbank Explorer

  - Java Reflection Explorer
  - JCA Explorer
  - Regeldatei Explorer
    → Siehe *XML-Eingangsnachricht für XML-EDI Adapter erstellen (Workbench/Process Engine: Modul-Guide, Kap. 7.3, S. 160).*
  - SAP Explorer
  - Web Services-Explorer
- **Speichern**:
  - **Datei**: Zum Speichern des angezeigten Inhalts.
  - **Datei als**: Öffnet einen Dateiexplorer zum Speichern des angezeigten Inhalts an beliebiger Stelle und in beliebigem Format.
  - **Repository**: Zum Speichern des angezeigten Inhalts im Repository.
  - **Zwischenablage**: Kopiert die angezeigten Inhalte in die Zwischenablage.
- **Drucken**: Öffnet den Drucker-Dialog zum Ausdrucken der aktuellen Datei.

Die folgenden Funktionen sind nicht in allen Bereichen verfügbar!

- **Geeignete Editoren für den Inhalt anzeigen**: Reduziert die verfügbaren Editoren auf diejenigen, welche für das aktuell angezeigte Dateiformat geeignet sind.
- **XML-Schema-Template erzeugen**: Erstellt aus einer XML-Datei eine Vorlage für ein XML Schema.
  → Siehe *XML Schema-Template aus XML-Datei erstellen (Workbench: Benutzer-Guide, Kap. 1.13.1.2, S. 56).*
- **Vergleichen:** Vergleicht eine XML-Datei mit einer Datei im Dateisystem, einer Datei im Repository oder mit dem Inhalt der Zwischenablage.
  → Siehe *XML-Dokumente vergleichen (Workbench: Benutzer-Guide, Kap. 1.12.3, S. 52).*
- **XPath-Suche**: Wenn markiert, dann kann im Suchen-Feld auch nach XPath-Ausdrücken gesucht werden.
  → Siehe *Mit XPath-Ausdrücken suchen (Workbench: Benutzer-Guide, Kap. 1.13.1.4, S. 57).*
- **Editierbar**: Wenn diese Option markiert ist, dann können Sie angezeigte Datei bearbeiten.
- **Parameter aus der Zwischenablage**: Kopiert Variablen aus der Zwischenablage in das XSLT-Stylesheet.

- **Toolbar an Content anpassen**: Abhängig vom Format des aktuellen Inhalts werden für den Bereich die passenden Befehle aktiviert. So sind z.B. bei ASCII Textdateien die Ansichten „Schema-Ansicht“ und „Baum-Ansicht“ nicht verfügbar.
- **Mehrfache Knoten ausblenden**: Blendet alle mehrfach vorhandenen Elemente aus
- **Daten aus der Struktur löschen**: Löscht alle Inhalte, die weder ein Element noch ein Attribut sind.
- **Nur Textknoten aus der Struktur löschen**: Löscht alle Inhalte von Elementen. Attributwerte bleiben erhalten.
- **Nur Attributtexte aus der Struktur löschen**: Löscht alle Werte von Attributen. Elementinhalte bleiben erhalten.

■ **XSLT Mapping Tabelle**

Ansicht, in der per Drag'n'Drop ein XSLT-Stylesheet erstellt werden kann. Wird nach dem Anlegen eines XSLT Converter Moduls automatisch angezeigt.

- Links: Anzeige von XPath-Ausdrücken und Werten
- Rechts: Elemente und Attribute
- Mittlere Spalte:
- Zelle mit Gleichheitszeichen: einem Attribut (rechte Spalte) wird ein Wert (linke Spalte) zugewiesen
- Zelle mit Pfeil: Zeigt an, dass einem Element (rechte Spalte) ein XPath-Ausdruck (linke Spalte) zugewiesen wird.

■ **Schema-Ansicht**

Zum Bearbeiten eines Schemas (nur verfügbar, wenn ein Schema geladen wurde)

→ Siehe *Schema-Editor (Workbench: Benutzer-Guide, Kap. 1.13.2, S. 57).*

■ **Baum-Ansicht**

■ **Text-Ansicht**

Zeigt die Elemente der Datei als Text an.

■ **Externen Editor öffnen**

Zeigt eine Liste aller Applikationsprofile an, die zum Öffnen der Datei verwendet werden können.

■ **Suche**: Siehe *Suchen in der inubit Workbench (Workbench: Benutzer-Guide, Kap. 1.9, S. 46).*

5.1.2 Weitere Funktionen

Drag'n'Drop

Der XSLT Mapper bietet folgende Möglichkeiten, Elemente, XPath-Ausdrücke oder Strukturen per Drag'n'Drop zu erstellen:

Der Bereich, in dem etwas erstellt werden soll, muss schreibbar sein. Wählen Sie dazu aus der Werkzeugleiste „ > Editierbar".

- Elemente kopieren zwischen allen Bereichen per Drag'n'Drop
- Elemente innerhalb eines Bereichs kopieren:
  Halten Sie beim Drag'n'Drop die Strg-Taste gedrückt.
- Neue Quell- oder Zielstruktur erstellen
  a. Wählen Sie aus der Werkzeugleiste „ > Neu > `<Typ_der_neuen_Datei>`".
  b. Ziehen Sie ein Element in den Bereich.
- XPath-Ausdruck erstellen
  Ziehen Sie ein Element aus der XML-Quell-Datei in die linke Spalte des Stylesheet-Bereichs.

Kontextmenüs

Im Stylesheet-, Quell- und Zielbereich können Sie über die Kontextmenüs der Elemente folgende Funktionen:

Funktion		Erläuterung
Kopieren		
Text ausrichten		Nur in den Textansichten verfügbar Verändert das Dokument
Ausschneiden		
Einfügen		
Knoten verschieben	nach links	Knoten vorher markieren
	nach rechts	
	nach oben	
	nach unten	
Knoten zuklappen		
Knoten aufklappen		
Alle untergeordneten Knoten aufklappen		
Eingabefeld für die Suche öffnen		
Menü unter dem Pfeil öffnen		Um Menü zu schließen, ohne einen Befehl auszuwählen Escape drücken.

Funktion	Erläuterung
Finde verwendete Knoten im Quelldokument	Wenn Sie ein Element in der Mapping-Tabelle markieren und diese Option aufrufen, werden alle Knoten in der XML-Quelldatei markiert, die für das Element benutzt werden. Wird kein Knoten in der XML-Quelldatei gefunden, wird dies mit einem Warnton angezeigt.
Unterbrechungspunkt an/aus	Zum Anlegen eines Unterbrechungspunktes. Sie können Ihr XSLT-Stylesheet während der Erstellung testen.
XSLT-Kommandoassistent:	Öffnet den XSLT Kommandoassistenten.
Skript testen	

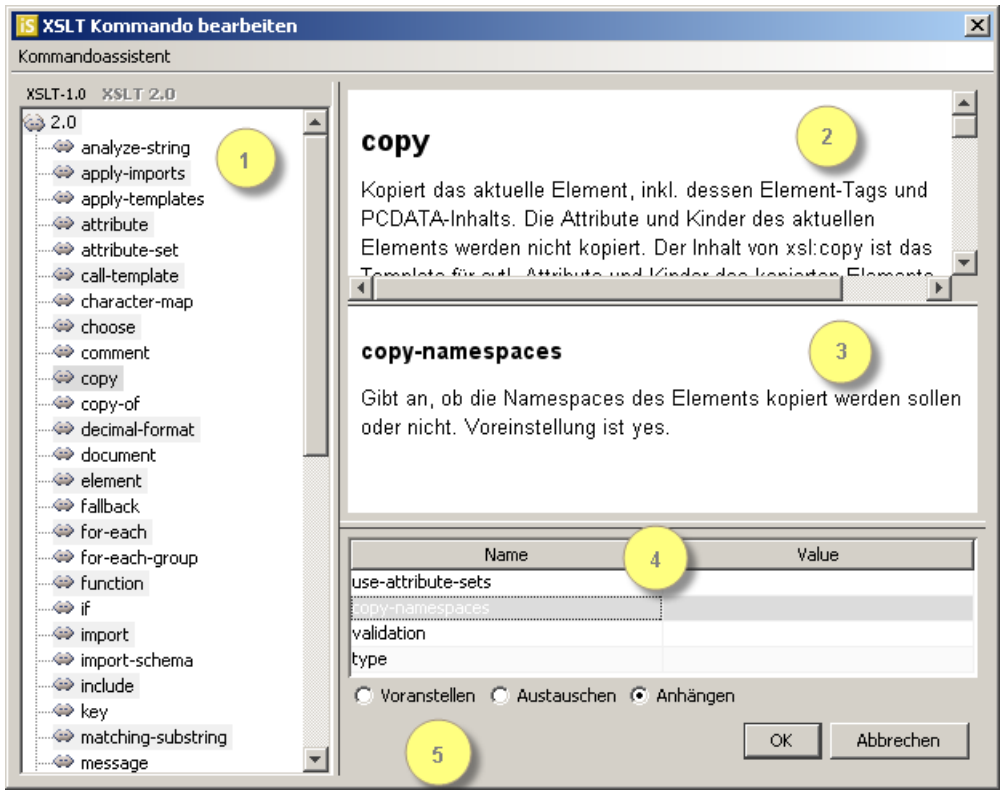
5.1.3 XSLT-Kommandoassistent

Aufruf

Führen Sie im Stylesheet-Bereich eine der folgenden Aktionen aus:

- In der rechten Spalte Element markieren > Kontextmenü öffnen > XSLT-Kommandoassistent
- In der mittleren Spalte oder doppelklicken.

Aus Zellen mit einem Gleichheitszeichen (Attribut mit Wertzuweisung) können Sie den Assistenten nicht aufrufen!



Der XSLT-Kommandoassistent gibt Ihnen Informationen zu dem XSLT-Element, das im Stylesheet-Bereich markiert ist, und fügt Elemente und Attribute an der gewünschten Position ein. Der XSLT-Kommandoassistent besteht aus folgenden Bereichen:

1. **Element-Liste**
   Wahlweise XSLT 1.0 oder 2.0.
2. **Element-Erläuterungen**
   Zu dem Element, das in der Liste markiert ist
3. **Attribut-Erläuterungen**
   Zu dem Attribut, das in der Attribut-Tabelle markiert ist.
4. **Attribut-Tabelle**
   Anzeige aller Attribute und Attributwerte des markierten Elements.
   - Schwarze Schrift: obligatorisches Attribut
   - Graue Schrift: optionales Attribut

   Die Wertebereiche der Attribute werden als Tooltipps über den Value-Feldern angezeigt. Um einem Attribut einen Wert zuzuweisen, klicken Sie in das Feld „Value“ und geben den Wert ein.
5. **Position des XSLT-Elements**
   - **Voranstellen**: Der Befehl wird auf die Hierarchieebene gestellt, auf der das aktuell markierte Element stand. Das Element wird darunter eingerückt.

- **Austauschen**: Wechselt den vorhandenen XSLT Befehl gegen den ausgewählten Befehl aus.
- **Anhängen**: Der Befehl wird unter das aktuell markierte Element und damit eine Hierarchieebene tiefer gestellt.

5.2 XSLT-Stylesheets testen

Sie können Ihr XSLT-Stylesheet bereits während der Erstellung testen.

So gehen Sie vor

Wählen Sie aus dem Menü „Bearbeiten > XSLT-Datei testen“ oder klicken Sie in der Toolbar auf .

Das Ergebnis und die benötige Ausführungszeit werden im Bereich „XML-Zieldatei“ im Register „Ergebnis des Mappings“ angezeigt.

5.3 XSLT-Stylesheets debuggen

Nutzen Sie die folgenden Buttons in der Werkzeugleiste

Button	Erläuterung
	Bis zum nächsten Breakpoint-Button. Führt das XSLT-Stylesheet bis zum nächsten Breakpoint aus (wenn vorhanden) oder bis zum Ende aus. Die aktuell ausgeführte Zeile und das damit korrespondierende Element in der XML-Quelldatei werden hervorgehoben. Im Bereich „XML-Ziel“ wird neben jedem XSLT-Elementen angezeigt, wie häufig das Element ausgeführt wurde. Das Ergebnis wird in der Text-Ansicht aufgebaut. Syntaktische Fehler werden sofort angezeigt.
	Trippeln-Button. Führt das Stylesheet zeilenweise bis zum nächsten Breakpoint (wenn vorhanden) oder bis zum Ende aus. Das Ergebnis wird schrittweise aufgebaut.

Button	Erläuterung
	Nächster Verarbeitungsschritt-Button. Führt die hervorgehobene Zeile aus.
	Pause-Button. Hält die Ausführung vorübergehend an. Um den Prozess an der Stelle fortzusetzen, an der Sie ihn gestoppt haben, klicken Sie erneut auf , oder .
	Stopp-Button. Beendet die Ausführung und entfernt die Infos zu der Element-Ausführung. Sie müssen die Ausführung erst beenden, damit das XSLT-Stylesheet wieder von der ersten Zeile an ausgeführt wird. Sonst wird die Ausführung immer dort fortgesetzt, wo sie zuletzt angehalten wurde.

5.4 Ergebnis validieren

Sie haben folgende Möglichkeiten, die Gültigkeit des Ergebnisses zu prüfen:

- **Gegen eine DTD validieren**
  Menü „Bearbeiten > DTD Prüfung der Ergebnisdatei“
- **Gegen ein XML-Schema validieren**
  Menü „Bearbeiten > Schema-Prüfung der Ergebnisdatei“

Der Pfad zum Schema bzw. zur DTD muss im XSLT-Stylesheet angegeben sein; wenn nicht, dann werden Sie aufgefordert, eine passende Datei zu öffnen.

Das Ergebnis der Validierung wird in einem eigenen Dialog angezeigt. Falls Fehler gefunden wurden, dann werden diese präzise angegeben, damit Sie Ihr Mapping entsprechend ändern können.

5.5 Extensions und Java-Packages verwenden

Dieser Abschnitt erläutert die folgenden Themen:

- *Extensions verwenden, S. 144*
- *Java-Packages der inubit Suite 6 einbinden, S. 145*

Überblick

Um in einem XSLT-Stylesheet Funktionalitäten zu realisieren, die nicht durch den XSLT-Befehlssatz abgedeckt werden, verwenden Sie Extensions oder Funktionen aus Java-Packages.

5.5.1 Extensions verwenden

Eine Vielzahl gebrauchsfertiger Extensions finden Sie auf folgenden Seiten:

- http://www.exslt.org
- http://xml.apache.org/xalan-j/extensionslib.html
- http://saxon.sourceforge.net

Sobald Sie Prozessor-spezifische Extensions einsetzen, ist Ihr XSLT-Stylesheet abhängig von dem ausgewählten XSLT-Prozessor!

So gehen Sie vor

1. Zeigen Sie Ihr XSLT-Stylesheet an.
2. Fügen Sie dem `xsl:stylesheet` Element über das Kontextmenü ein `namespace`-Attribut für Ihre Extension hinzu, z. B.:
   ```
   <xsl:stylesheet
   xmlns:string="http://exslt.org/string"
   ```
3. Fügen Sie das folgende Attribut hinzu:
   ```
   extension-element-prefixes="string"
   ```

Extensions aufrufen

Eine Extension-Funktion wird über den festgelegten Namensraum und ihren Namen aufgerufen, z. B.:

```
<xsl:template match="/">
  <tokens>
    <xsl:for-each
      select="string:tokenize('A,B,C,D,' , ',' )">
      <xsl:copy-of select="."/>
    </xsl:for-each>
```

5.5.2 Java-Packages der inubit Suite 6 einbinden

Beschreibungen der inubit Suite 6-eigenen Java-Packages finden Sie im Verzeichnis `<iS-installdir>/pluginSDK/apidoc/index.html`,unter `com.inubit.ibis.xsltext.Misc` und `com.inubit.ibis.xsltext.Formatter`.
Voraussetzung: Sie haben beim Installieren der inubit Suite 6 das SDK installiert.

So gehen Sie vor

1. Zeigen Sie Ihr XSLT-Stylesheet an.
2. Fügen Sie dem `xsl:stylesheet` Element über das Kontextmenü ein `namespace`-Attribut hinzu. Geben Sie als Wert den Package-Namen an, z. B.:

```
<xsl:stylesheet
xmlns:Misc="com.inubit.ibis.utils.xsltext.Misc“>
```

Funktion aus Package aufrufen

Eine Methode aus einem Package wird mit dem festgelegten Präfix und Namen aufgerufen:

```
<xsl:template match="/">
  <DateTime>
    <xsl:value-of
    select="Misc:getDateTime('yyyyMMddHHmmss')"
    />
  </DateTime>
```

5.6 Beliebige XSLT-Prozessoren einbinden

Für die Transformationssprache XSLT können verschiedene Prozessoren eingesetzt werden, gängig sind z. B. die mitgelieferten Prozessoren Xalan und SAXON.

Neben den mitgelieferten XSLT-Prozessoren können Sie beliebige andere XSLT-Prozessoren verwenden. Dazu benötigen Sie eine JAR-Datei mit dem XSLT Prozessor, dieses muss dem dafür festgelegten Java Interface entsprechen.

So gehen Sie vor

1. Zeigen Sie in der inubit Workbench das Register „Configuration“ an.

2. Um die JAR-Datei auf die inubit Process Engine zu laden, wählen Sie im Menü „Configuration > Bibliothek hochladen“.
3. Öffnen Sie die Eigenschaften Ihres XSLT Converters zum Bearbeiten oder erzeugen Sie ein neues Modul.
4. Geben Sie im *Dialog „XSLT Converter Eigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 5.8, S. 148)* im Feld „XSLT Prozessor“ den Klassennamen des neuen XSLT-Prozessors ein.
5. Klicken Sie auf „Fertigstellen“.

Nun können Sie den neuen XSLT-Prozessor verwenden.

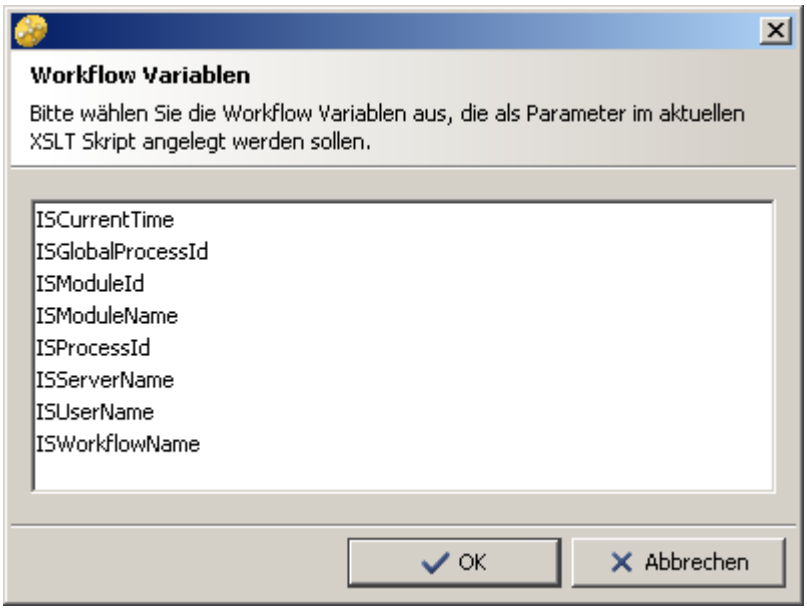
5.7 Variablen verwenden

Sie können im XSLT Converter auf Variablen zugreifen, um z. B. verständliche Fehlermeldungen an einen Fehlerausgang zu senden.

Beachten Sie beim Einsatz von Variablen mit dem Typ `base64Binary` in XSLT Convertern aus iS-Versionen 5.1 und älter: Der Variablen-Inhalt wurde bei diesen XSLT Convertern komprimiert ausgegeben. Daher müssen Sie zum Dekodieren des Variablen-Inhalts die XPath-Funktion `misc:decodeWithDecompression($variable)` verwenden. Alternativ legen Sie einen neuen XSLT Converter an. Bei diesem können Sie direkt auf den unkomprimierten Variablen-Inhalt zugreifen.

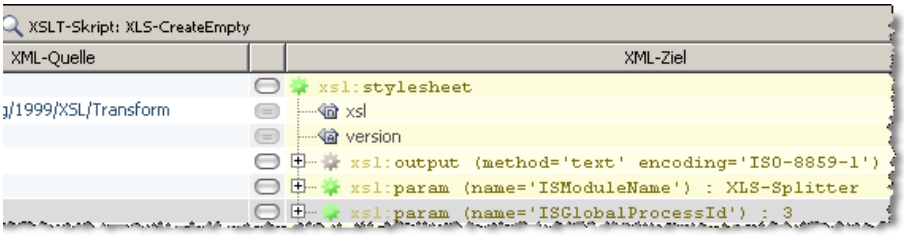
So gehen Sie vor

1. Erstellen Sie ein XSL Converter Modul und integrieren Sie es in einen Technical Workflow.
2. Testen Sie den Technical Workflow.
3. Doppelklicken Sie den Watchpoint vor dem XSLT Converter Modul. Ein Dialog öffnet sich und zeigt die Ergebnisdatei mit allen Variablen an:
4. Um die Ergebnisdatei mit den Variablen in die Zwischenablage zu kopieren, klicken Sie auf „Daten kopieren“. Schließen Sie den Dialog.
5. Öffnen Sie das XSLT Converter Modul zum Bearbeiten.
6. Öffnen Sie im Stylesheet-Bereich das Menü ( ) und wählen Sie „Parameter aus der Zwischenablage“. Ein Fenster zum Auswählen einer Variablen öffnet sich, z. B.:



7. Markieren Sie eine Variable und klicken Sie auf „OK“.

Das Fenster schließt sich. Aus der Variablen wird ein gleichnamiger Parameter erstellt und im Stylesheet-Bereich angezeigt:



Wenn der Parameter bei der Ausführung des Technical Workflows nicht von außen gesetzt wird, dann wird der hier angezeigte Wert als Default übertragen. Daher sollten Sie für Parameter immer Standardwerte vergeben.

→ Siehe

- *Workflow-Variablen und Mappings (Workbench: Benutzer-Guide, Kap. 14, S. 365)*
- *Test-Modus: Diagramme testen (Workbench: Benutzer-Guide, Kap. 16.1, S. 420)*

5.8 Dialog „XSLT Converter Eigenschaften“

Transformator Eigenschaften

- **XSLT Prozessor**
  Zum Auswählen eines XSLT-Prozessors. Die XSLT-Prozessoren unterscheiden sich hinsichtlich des von ihnen unterstützten Sprachumfangs.
- **Prozessor-Attribute**
  In dieser Tabelle geben Sie Eigenschaften mit Werten für den XSLT-Prozessor an (implementierungsspezifische Transformer-Factory-Attribute).

Mapper Eigenschaften

- **Quell- und Zieldateien in der Process Engine speichern**
  Wenn markiert, dann werden die Quell- und Zieldateien auf der inubit Process Engine gespeichert und können somit auch auf anderen Client-Rechnern bearbeitet werden.

Wenn die Quell- und Zieldateien auf der inubit Process Engine sehr groß sind, kann die Performance beeinträchtig werden, weil die Dateien bei jeder Ausführung des XSLT Converters geladen werden.

Wenn Sie das Modul bereits publiziert haben und später diese Option abwählen, dann werden die Dateien von der inubit Process Engine und die Referenz zur lokalen Kopie auf dem Client-Rechner gelöscht.

- **Eingangsnachricht ignorieren**
  Wenn die Option markiert ist, dann wird der Bereich „XML-Quelldatei“ im XSLT Mapper ausgeblendet.
  Diese Option ist sinnvoll, um mit einem oder mehreren XSLT-Stylesheets strukturierte Ausgangsnachrichten neu zu erstellen.
- **XSLT Stylesheet cachen**
  Wenn der XSLT Converter mehrfach kurz hintereinander aufgerufen wird, dann verbessert das Cachen des Stylesheets die Performanz.

→ Siehe
- *XSL-Versionen und EXSL-Unterstützung (Workbench/Process Engine: Modul-Guide, Kap. 5, S. 133)*
- *Beliebige XSLT-Prozessoren einbinden (Workbench/Process Engine: Modul-Guide, Kap. 5.6, S. 145)*

Dieser Abschnitt erläutert die folgenden Themen:

- *Beispiel: Konvertierung CSV nach XML, S. 150*
- *Register „Adapter Editor“, S. 151*

Verwendung

Der CSV Adapter konvertiert eingehende CSV-Nachrichten in XML-Nachrichten bzw. XML nach CSV:

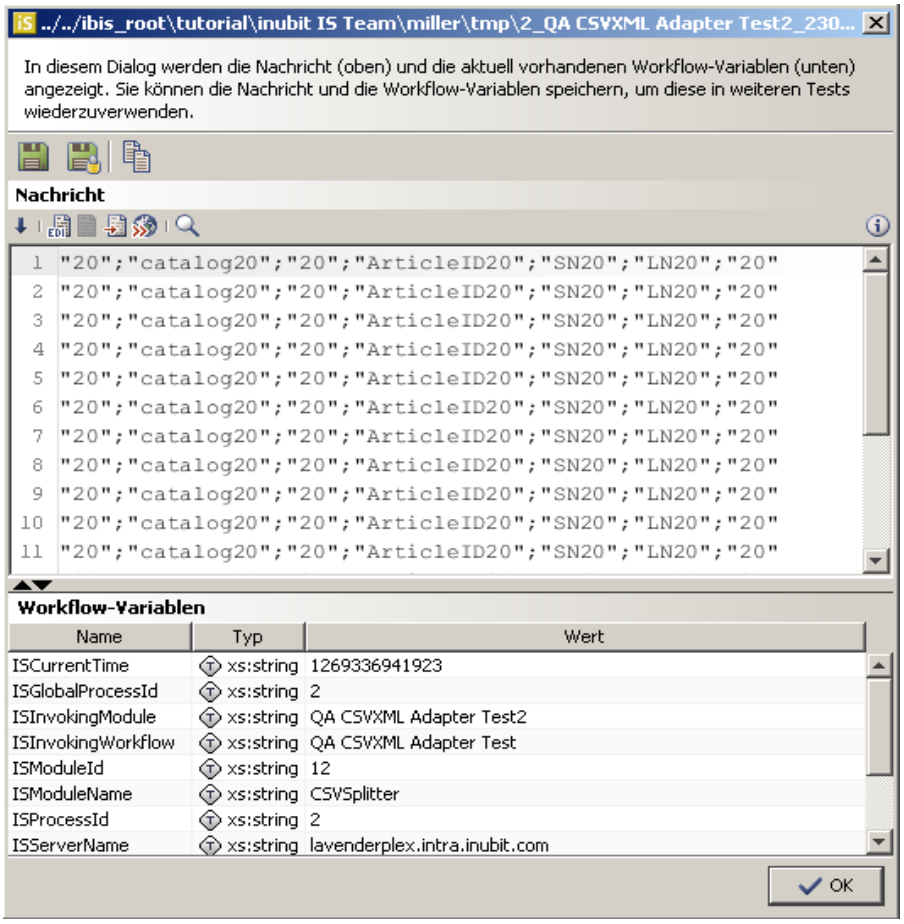
- **CSV > XML**

  Sinnvoll, um Nachrichten im weiteren Ablauf eines Workflows per XSLT in ein Standard-XML-Format wie z. B. xCBL oder BMEcat konvertiert zu können.
- **XML > CSV**

  Konvertiert Daten aus einem XML-Format in ein CSV-Format. Als XML-Eingabeformat wird ein „flaches“ XML-Format erwartet, das die einzelnen, zu transformierenden Datensätze enthält. Dieses XML-Format müssen Sie mit Hilfe eines XSLT Converter Moduls erzeugen.

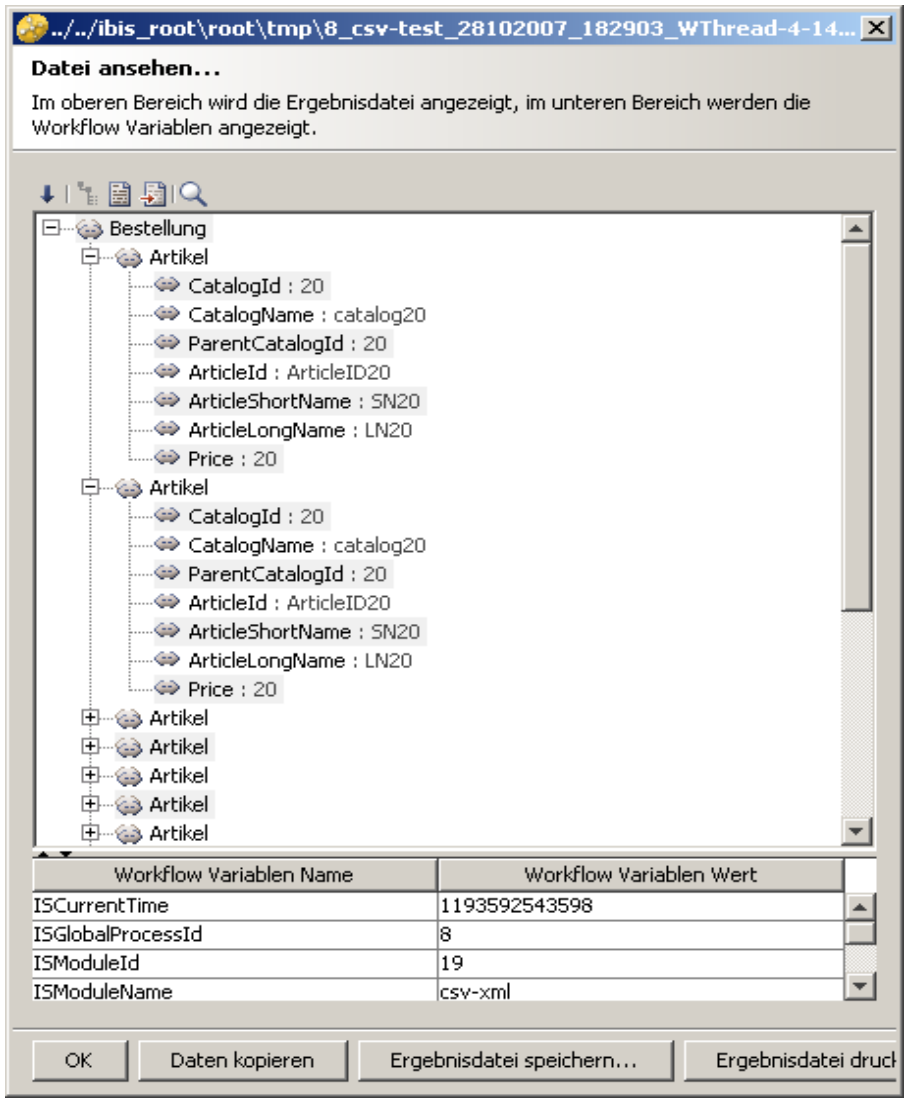
6.1 Beispiel: Konvertierung CSV nach XML

Eingangsnachricht



Der Adapter entfernt aus den einzelnen Feldern der CSV-Datei Leer- und/oder Steuerzeichen vor und nach dem String, bevor er die Werte in die XML-Elemente der Ausgangsdatei übernimmt. Dadurch verhindert er, dass bei der Verarbeitung von CSV-Daten, deren Spalten feste Längen haben und z. B. mit Leerzeichen ausgefüllt werden, Problem mit der daraus erzeugten XML-Datei auftreten.

Ausgangsnachricht



6.2 Register „Adapter Editor“

- **XML Root Tag**
  Bezeichner für das Wurzel-Element des zu erzeugenden XML-Dokuments, z. B. `<AUFTRAG>`.
- **Segmentname**
  Der Segmentname wird in das XML-Tag des mit dem Format Adapter erzeugten XML-Dokuments übernommen, das die einzelnen Attribut-Tags umschließt, z.B. `<AUFTRAGSPOSITION>`.
- **Textkennzeichner**

Zeichenkette, welche die einzelnen Werte in einem Datensatz der Quellnachricht umschließt und als Text kennzeichnet.

- **Feldnamen**
  In manchen CSV-Formaten befinden sich die Attributnamen der einzelnen Felder in der ersten Zeile jeder CSV-Datei. Falls das in Ihrem CSV-Format der Fall ist, markieren Sie die Auswahlbox.
- **Überstehende Felder ignorieren/Fehlende Felder ignorieren**
  (nur bei CSV>XML)
  Ermöglichen das Parsen „nicht-CSV-konformer“ Nachrichten.
- **Spalten mit fester Breite**
  (nur bei XML>CSV)
  Für jeden Feldtyp wird dessen maximale Länge ermittelt. Alle anderen Felder des gleichen Typs werden mit Leerzeichen aufgefüllt, bis die maximale Länge erreicht ist.
- **Feldbegrenzer**
  Zeichenkette, welche die einzelnen Werte in einem Datensatz der Quellnachricht voneinander abgrenzt.
- **Datensatzbegrenzer**
  (Nur bei XML-CSV Konvertierung)
  Bei CSV-XML Konvertierung ist \n als Datensatzbegrenzer zu verwenden. Wird das Zeichen für Datensatzbegrenzer in den Daten verwendet, muss es durch einen Backslash (\) escaped bzw. durch zwei Ausrufezeichen (`!!`) umschlossen werden.
- **Datensatzbegrenzer am Dateiende**
  Nur bei einem XML-CSV Adapter auswählbar.
- **Zeichensatzkodierung** (CSV)
  Gibt an, in welchem Zeichensatz die Eingangsnachricht kodiert ist. Der Standard ist der Zeichensatz ISO-8859-1, der alle Zeichen der gebräuchlichen westeuropäischen Sprachen inkl. der Sonderzeichen enthält. Sie können einen anderen Zeichensatz auswählen oder eingeben.
- **Feldnamen aus CSV Datei**
  Statt die Feldnamen aus der Eingangsnachricht einzulesen, können diese auch aus einer beliebigen, anderen CSV-Datei gelesen werden.

Denken Sie daran, die Feld- und Datensatzbegrenzer so zu definieren, wie sie in der Datei vorkommen, die importiert werden soll.

- **Feldnamen aus XML Datei**
  Statt die Feldnamen aus der Eingangsnachricht einzulesen, können diese auch aus einer XML-Datei gelesen werden. In diesem Fall müssen die Angaben in den Feldern „XML Root Tag“ und „Segmentname“ mit den Namen der Root- und Segment-Elemente in der Datei übereinstimmen, die Sie importieren wollen.

- **Tabellen-Anzeige**
  Zeigt die Feldnamen aus der importierten Datei an. Sie können weitere Zeilen in die Tabellen einfügen und löschen bzw. angeben, dass eine Zeile bei der Auswertung übergangen werden soll. Verwenden Sie dazu das Kontextmenü der Tabelle.

Dieser Abschnitt erläutert die folgenden Themen:

- *Funktionsprinzip der EDI>XML-Konvertierung, S. 156*
- *EDI-XML Adapter erstellen und Regel konfigurieren, S. 158*
- *XML-Eingangsnachricht für XML-EDI Adapter erstellen, S. 160*
- *Neue Regel erstellen, S. 161*
- *Regeln zur Wiederverwendung auf die inubit Process Engine publizieren, S. 162*
- *SAP IDoc XML-Regel erstellen, S. 163*
- *Regeleditoren verwenden, S. 164*
- *Dialogbeschreibungen, S. 166*

Verwendung

Mit dem EDI Format Adapter werden EDI-Nachrichten regelbasiert konvertiert. EDI-Nachrichten ermöglichen die automatische Weiterverarbeitung der Geschäftskorrespondenz im Softwaresystem des Empfängers ohne manuellen Eingriff.

Für gängige EDI-Standards wie z. B. UN/EDIFACT, ANSI X12 oder VDA sind Konvertierungsregeln vorhanden und können sofort eingesetzt werden. Falls für Ihre EDI-Nachrichten in der inubit Suite 6 keine passenden Regeln vorhanden sind, unterstützt Sie der EDI Regeleditor beim Erstellen einer neuen Regel.

Die Nutzung der EDI-Regeln ist lizenzpflichtig: Es werden immer nur die EDI-Standards angezeigt, für deren Regeln Sie eine Lizenz erworben haben!

Adaptertypen

Für die Konvertierung von EDI-Nachrichten gibt es zwei Adaptertypen:

- **EDI-XML Adapter**: Konvertiert EDI-Nachrichten in ein EDI-XML Format.
- **XML-EDI Adapter**: Konvertiert EDI-XML Nachrichten in EDI-Nachrichten.

HL7-Daten via TCP/IP MLLP:
Für die Übertragung und Bestätigung von HL7-Nachrichten über TCP/IP MLLP nutzten Sie den HL7 TCP/IP Connector. Siehe *HL 7 TCP/IP Connector (Workbench/Process Engine: Systemkonnektor-Guide, Kap. 12, S. 143).*

7.1 Funktionsprinzip der EDI>XML-Konvertierung

Für einen EDI Adapter benötigen Sie immer zwei Dateien:

1. **EDI-Eingangsnachricht**
   Beispielnachricht, die Sie von Ihrem Geschäftspartner erhalten oder die Sie an Ihren Geschäftspartner senden
2. **Konvertierungsregel**
   XML-Datei mit Regeln für die Konvertierung. Regeln können grundsätzlich für beide Konvertierungsrichtungen verwendet werden.
   Die Regeldatei kann im Adapter gespeichert werden oder (bei der Autoerkennung von Nachrichten) auf der inubit Process Engine.
   Bei der Konvertierung wird automatisch die Syntax der Nachrichten geprüft.

Die folgende Abbildung illustriert den Zusammenhang zwischen einer EDI-Eingangsnachricht und der Konvertierungsregel:

EDI-Eingangsnachricht

Damit EDI-Nachrichten vom Empfänger verarbeitet werden können, müssen die Zeichen in der Nachricht bezüglich ihrer Syntax und Semantik einer bekannten Struktur entsprechen. Diese Strukturen sind in den verschiedenen EDI-Standards festgelegt.

Prinzipiell besteht eine EDI-Nachricht aus Metainformationen und dem Payload (Nutzinhalt):

- Metainformationen enthalten Information über die Verarbeitung, speziell über das Routen an den Empfänger. In der Abbildung ist das z. B. das EDI-Element `UNH`, das dem XML-Element `MessageHeader` entspricht.
- Der Payload sind die Daten, die das Empfängersystem verbucht. In der abgebildeten Nachricht beginnt der Payload mit dem EDI-

Element `BGM` (entsprechend dem XML-Element `BeginningOfMessage`).

Nutzinhalt und Metainformationen bestehen aus Elementen, die fortlaufend zu Segmenten zusammengesetzt sind:

- Für jedes Element sind Eigenschaften wie Länge, Position im Segment, Typ (numerisch/alphanumerisch), optionales oder obligatorisches Auftreten und Trennzeichen vorgegeben.
- Für alle Segmente ist vorgegeben, in welcher Reihenfolge diese wie oft auftreten dürfen oder müssen.

Konvertierungsregel

Die Regel gibt auf Basis des Standards vor, wie EDI-Elemente bzw. EDI-Segmente auf XML-Elementen abgebildet werden. Die Abbildung zeigt dieselbe Konvertierungsregel im EDI-XML Adapter in zwei Ansichten:

- **Ansicht „Attribut-ID“**
  Zeigt von der Regeldatei alle Attribute mit dem Namen „ID“ an. Diese ID entspricht dem Bezeichner des EDI-Elementes bzw. -Segmentes. Die Bezeichner in der Nachricht müssen den IDs in der Regeldatei entsprechen, sonst wird die Nachricht nicht verarbeitet.
  Nach der ID zeigt der Buchstabe `C` bzw. `M` an, ob das Element optional (engl. conditional) oder obligatorisch (engl. mandatory) ist. Die Ziffern in eckigen Klammern geben an, wie häufig das Element in der Nachrichtenstruktur vorkommen darf.
  In der Abbildung ist zu sehen, dass in der EDI-Nachricht das Element CNT dreimal erscheint. Damit ist das Element regelkonform, denn in der Regel ist angegeben, dass das Element gar nicht oder bis zu neunmal vorkommen darf: `CNT C[0…9]`.
- **Ansicht „XML-Tag“**
  Zeigt statt der EDI-IDs die XML-Tags an, auf welche die IDs abgebildet werden. Beispiel: Das Element `CNT` erhält den XML-Tagnamen `ControlTotal`.

„Sprechende“ XML-Tagnamen verbessern in der Regel die Lesbarkeit von EDI-XML-Nachrichten und erleichtern deren Verarbeitung im Workflow z. B. bei nachfolgenden XSLT-Konvertierungen. Um Tagnamen auszugeben, setzen Sie in den Moduleigenschaften die Option „Ausgabeformat der XML-Tags" auf „xmlTag“.
Beachten Sie, dass lange XML-Tagnamen große EDI-XML-Strukturen erzeugen, welche die Performance beeinträchtigen. Wenn die Performance relevant ist, vergeben Sie kurze XML-Tagnamen (max. 3 Zeichen) oder setzen Sie in den

Moduleigenschaften die Option „Ausgabeformat der XML-Tags" auf „id“.

→ Siehe *Namen der XML-Elemente (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 169)*.

7.2 EDI-XML Adapter erstellen und Regel konfigurieren

Voraussetzungen

- Sie kennen den Standard und die Version des Standards, zu dem die Nachrichten konform sind, z. B. EDIFACT, IFCSUM, Version von 1996, Release A.

  Wenn die Nachrichten lediglich ähnlich zu einem Standard sind oder keinem Standard entsprechen, dann benötigen Sie genaue Kenntnis über den Aufbau der Nachricht (Segmente, Elemente).
- Sie kennen die verwendeten Trennzeichen.

So gehen Sie vor

1. Erzeugen Sie einen EDI Adapter:
   - Wenn die Nachricht einem der Formate EDIFACT, ANSI X12, HL7 oder einem Subset davon, z. B. EDIGAS, entspricht, dann aktivieren Sie die automatische Regelerkennung im *Dialog „EDI/XML-Adaptereigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 7.8.2, S. 168)*. Damit wird das Erstellen einer Regeldatei am Modul überflüssig.

     Diese Option ist für alle EDIFACT, ANSI X12 und HL7-Nachrichten verfügbar, für die eine passende Regel auf der inubit Process Engine vorhanden ist.

     Nach dem Beenden des Assistenten können Sie den EDI Adapter sofort einsetzen.
   - Wenn die Nachricht in einem anderen Format vorliegt oder keinem Standard-Format entspricht, dann lassen Sie sich vom Assistenten durch die Erstellung des Adapters leiten.

Alternativ können Sie Regeln von der Process Engine herunterladen: Öffnen Sie dazu das ⬇ -Menü und wählen Sie „Von der Process Engine herunterladen“. Verwenden Sie diesen Befehl, um eine Regel als Vorlage zu verwenden, die bereits auf die inubit Process Engine publiziert wurde. Im Auslieferungsumfang sind bereits mehrere Standards enthalten. Die im Auslieferungsumfang enthaltenen EDIFACT-Regeln

enthalten keine Qualifierlisten. Sie können die Qualifier selbst definieren oder die Regel aus der Bibliothek neu erzeugen.

2. Zum Erstellen einer neuen Regel öffnen Sie im Register „EDI-XML Adapter“ das ⬇ -Menü und wählen „Neu > Regel“. Der Regel-Assistent öffnet sich.
3. Wählen Sie anhand des Nachrichtentyps die entsprechende Option:
   - **Regel benutzerdefiniert erstellen**: Wenn keine passende oder ähnliche Regel vorhanden ist. Es wird eine Regeldatei erzeugt, die nur das Root-Element `Message` enthält. Als Vorlage können Sie zusätzlich einen Standard auswählen, um die Strukturinformationen im Feld „Struktur“ automatisch setzen zu lassen. Falls dieser Wert nicht geeignet ist, dann müssen Sie als Standard den Wert „Custom“ wählen und die Struktur manuell setzen.
     → Siehe *Neue Regel erstellen (Workbench/Process Engine: Modul-Guide, Kap. 7.4, S. 161).*
   - **Regel aus Bibliothek erzeugen:** Für die Standards EDIFACT, ANSI X12 und HL7 existieren vollständige Regeln, die Sie als Vorlage benutzen können, um diese an Ihr spezielles Format anzupassen.
   - **Regel aus XML Schema erzeugen:** Für SAP IDoc können Sie ein entsprechendes Schema als Vorlage laden.
4. Lassen Sie sich von dem Regel-Assistenten durch die Erstellung führen. Für die aus einer Bibliothek erzeugte Regel wählen Sie die Version aus. Für die aus einem XML Schema erzeugte Regel sind keine weiteren Angaben erforderlich.

Nach Beenden des Assistenten können Sie im Regeleditor die komplette Regelstruktur der neuen Regel manuell bearbeiten.

→ Siehe *Regeleditoren verwenden (Workbench/Process Engine: Modul-Guide, Kap. 7.7, S. 164).*

Sie können selbst Regeln auf die inubit Process Engine publizieren und damit anderen zur Verfügung stellen. Siehe *Regeln zur Wiederverwendung auf die inubit Process Engine publizieren (Workbench/Process Engine: Modul-Guide, Kap. 7.5, S. 162).*

7.3 XML-Eingangsnachricht für XML-EDI Adapter erstellen

Der XML-EDI Adapter erwartet Eingangsnachrichten in einem bestimmten XML-Format, das von dem gewünschten EDI-Zielformat abhängig ist. Dieser Abschnitt beschreibt, wie Sie die erwartete XML-Eingangsnachricht erstellen.

Voraussetzungen

- Der XML-EDI Adapter ist bereits angelegt und wie folgt konfiguriert:
  - Die Auto-Erkennung ist **nicht** aktiviert.
  - Die Regeldatei ist am XML-EDI Adapter hinterlegt.
- Der XML-EDI Adapter ist bereits publiziert.
- Sie verfügen über eine Beispiel-Eingangsnachricht im XML-Format.

So gehen Sie vor

1. Erstellen Sie einen XSLT Converter.
2. Öffnen Sie im Bereich „XML-Zieldatei“ das -Menü und wählen Sie „Öffnen > Regeldatei Explorer“. Der Dialog „Regeldatei Explorer“ wird angezeigt.
3. Wählen Sie Ihren XML-EDI Adapter aus der Liste aus.
4. Optional:
   Markieren Sie die Optionen, um weitere Informationen über die Regelstruktur zu erhalten. Die Informationen werden als Kommentare im XML-Template angezeigt.
5. Klicken Sie auf „Fertig stellen“, um den Dialog zu schließen.
   Auf Basis der Regeldatei, die im Adapter hinterlegt ist, wird die Struktur der XML-Eingangsnachricht am ausgewählten XML-EDI Adapter erzeugt und angezeigt.
6. Ziehen Sie das Root-Element der Struktur nach oben auf das `xsl:template`-Element, um die Struktur in das Mapping zu übernehmen.
7. Laden Sie Ihre Beispiel-Eingangsnachricht in den Bereich „XML-Quelldatei“.
8. Füllen Sie die Struktur, in dem Sie Elemente aus Ihrer XML-Eingangsnachricht per Drag‘n‘Drop auf die Strukturelemente abbilden.

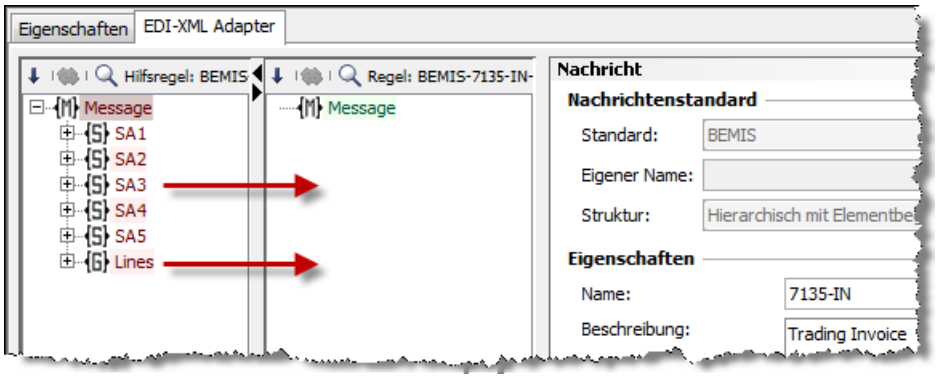
7.4 Neue Regel erstellen

Falls es keine Regeln gibt, die Sie für Ihre Zwecke anpassen können, erzeugen Sie eine leere Regel und fügen die Elemente der neuen Regel manuell ein.

Für dieses Vorgehen benötigen Sie die Spezifikation des Standards!

So gehen Sie vor

1. Zeigen Sie einen Regeleditor an.
2. Klicken Sie auf ⬇ und wählen Sie „Neu > Regel…“. Der *Assistent für neue Regel: Erstellungsmethode auswählen (Workbench/Process Engine: Modul-Guide, Kap. 7.8.6, S. 174)* öffnet sich.
3. Wählen Sie „Regel benutzerdefiniert erstellen > Neue Regel“.
4. Klicken Sie auf „Weiter“. Der *Assistent für neue Regel: Neue Regel (Workbench/Process Engine: Modul-Guide, Kap. 7.8.7, S. 174)* wird angezeigt. Füllen Sie die Felder aus.
5. Klicken Sie auf „Fertig stellen“.
   Im Regeleditor wird im Zielfenster eine leere Regel angezeigt, diese Regel enthält lediglich das Root-Element „Message“.
6. Wenn die Struktur der neuen Regel ähnlich der Struktur einer bereits vorhandenen Regel ist, laden Sie die vorhandene Regel als Hilfsregel: Klicken Sie auf ⬇ und wählen Sie „Hilfsregel-Ansicht“. Der Bereich „Hilfsregel“ wird angezeigt.
7. Klicken Sie im Bereich „Hilfsregel“ auf ⬇ und wählen Sie „Öffnen“ oder „Von der Process Engine herunterladen“. Die gewählte Hilfsregel wird angezeigt (rot hinterlegt).
8. Ziehen Sie die gesamte Regel, ausgewählte Segmente oder Elemente in das Zielfenster, um die neue Regel schrittweise aufzubauen:



9. Nutzen Sie im Zielfenster das Kontextmenü, um Segmente, Segmentgruppen oder Elemente hinzuzufügen, zu löschen oder zu verschieben.

10. Füllen Sie zu jedem Bestandteil der Regel die Felder im rechten Eigenschaftsfenster aus.

Abhängig vom verwendeten Regeleditor können Sie nach dem Bearbeiten der Regel den EDI Adapter publizieren oder die Regel auf die inubit Process Engine publizieren.

→ Siehe *Regeleditoren verwenden (Workbench/Process Engine: Modul-Guide, Kap. 7.7, S. 164).*

Wenn Sie eine Hilfsregel geöffnet haben und eine neue Regel mit einer anderen Struktur laden, wird eine Fehlermeldung ausgegeben. Sie können den Vorgang abbrechen, ansonsten wird die bisherige Hilfsregel entfernt. Das Öffnen einer Hilfsregel mit einer von der Regel abweichenden Struktur wird ebenfalls verhindert.

7.5 Regeln zur Wiederverwendung auf die inubit Process Engine publizieren

Einmal erzeugte Regeln können in verschiedenen EDI Adaptern wiederverwendet werden. Dazu müssen Sie die Regeln auf die inubit Process Engine publizieren.

Anschließend können Sie in jedem EDI Adapter auf diese Regeln zugreifen und eine Regel als Vorlage laden.

Voraussetzungen

Die Regel ist bereits an einem EDI Adapter vorhanden.

So gehen Sie vor

1. Zeigen Sie den EDI Adapter im lokalen Modus an.
2. Öffnen Sie das ⬇ -Menü und wählen Sie „Speichern unter“, um die Regel temporär im Dateisystem abzulegen.
3. Wählen Sie „Extras > EDI-Regeleditor…“, um den EDI-Regeleditor zu öffnen.
4. Öffnen Sie das ⬇ -Menü und wählen Sie „Öffnen > Datei…“, um die Regel aus dem Dateisystem zu laden. Die Regel wird angezeigt.
5. Öffnen Sie das ⬇ -Menü und wählen Sie „Auf die Process Engine publizieren“.

Die Regel wird publiziert. Dabei wird automatisch ein Dateiname für die Regel generiert. Die Werte für den generierten Dateinamen stammen aus den Angaben, die im Root-Element der Nachricht

(Element `Message`) gemacht wurden. Falls bereits eine gleichnamige Datei auf der inubit Process Engine vorhanden ist, dann wird diese überschrieben.

Ändern Sie den generierten Dateinamen nicht! Wenn der Dateiname nicht der Konvention entspricht, kann die Regel nicht mehr für die Autoerkennung verwendet werden. Der Name muss folgendes Format haben: `<Nachrichtentyp>-<Nachrichtenname>-<Version>-<Release>.xml`
Falls Release oder die Version nicht bekannt sind, geben Sie dafür ein Minuszeichen an, z. B. `EDIFACT-ORDERS---96A.xml`. Alle Buchstaben des Dateinamens müssen Großbuchstaben sein.

7.6 SAP IDoc XML-Regel erstellen

Voraussetzungen

Sie benötigen eine SAP IDoc Schema-Datei, die auf einem Rechner gespeichert sein muss, den Sie über Ihr Netzwerk erreichen können!

Das SAP Interface Repository, aus dem früher Schemas geladen werden konnten, wurde Ende 2006 von SAP komplett eingestellt; bisher wurde kein Ersatz zur Verfügung gestellt!

So erstellen Sie eine Regel nach SAP IDoc Standards:

1. Zeigen Sie einen Regeleditor an.
2. Klicken Sie auf ⬇ und wählen Sie „Neu > Regel“. Der *Assistent für neue Regel: Erstellungsmethode auswählen (Workbench/ Process Engine: Modul-Guide, Kap. 7.8.6, S. 174)* öffnet sich.
3. Wählen Sie „SAP IDoc“.
4. Geben Sie Pfad und Namen der SAP IDoc Schema-Datei ein oder klicken Sie auf „Datei suchen“, um zu der Schema-Datei zu navigieren.
5. Klicken Sie auf „Fertig stellen“.

Die Regel wird aus dem Schema erstellt und angezeigt. Abhängig vom verwendeten Regeleditor können Sie nach dem Bearbeiten der Regel den EDI Adapter publizieren oder die Regel auf die inubit Process Engine publizieren.

→ Siehe *Regeleditoren verwenden (Workbench/Process Engine: Modul-Guide, Kap. 7.7, S. 164).*

7.7 Regeleditoren verwenden

In den Regeleditoren zeigen Sie Regeln zum Konvertieren von EDI-Nachrichten an, bearbeiten diese und erstellen neue Regeln.

In der inubit Workbench gibt es folgende Regeleditoren:

1. **EDI-Regeleditor**
   Diesen Editor verwenden Sie, um Regel-Dateien auf der inubit Process Engine zu bearbeiten, neue Regeln zu erstellen und auf die inubit Process Engine zu publizieren.
   → Siehe *Regeln zur Wiederverwendung auf die inubit Process Engine publizieren (Workbench/Process Engine: Modul-Guide, Kap. 7.5, S. 162).*
   Die Regeln werden zur Autoerkennung von Nachrichten verwendet und stehen als Vorlagen allen inubit Workbench-Benutzern zur Verfügung, die auf EDI Adapter zugreifen können.
2. **Register „XML-EDI Adapter“ bzw. „EDI-XML Adapter“ eines lokalen Adapters**
   In diesem Editor erzeugen und bearbeiten Sie eine Regel, die im Adapter gespeichert wird.
   Diese Regel kann nicht von anderen Adaptern wiederverwendet und auch nicht zur Autoerkennung von Nachrichten genutzt werden.

Wenn die Option „Autoerkennung für EDIFACT, X12 und HL7“ im *Dialog „EDI/XML-Adaptereigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 168)* markiert ist, dann wird ein leeres Register mit einem entsprechenden Hinweis angezeigt.

Bearbeiten

Um eine Regel zu erweitern und zu bearbeiten, markieren Sie ein Element und öffnen das Kontextmenü.

Menüleiste

In der Menüleiste des Editors gibt es folgende Befehle:

- : Öffnet ein Untermenü mit folgenden Befehlen:
  - **Neu > Regel…**: Öffnet den *Assistent für neue Regel: Erstellungsmethode auswählen (Workbench/Process Engine: Modul-Guide, Kap. 7.8.6, S. 174)*.
    → Siehe *EDI-XML Adapter erstellen und Regel konfigurieren (Workbench/Process Engine: Modul-Guide, Kap. 7.2, S. 158).*
  - **Öffnen**
    Zum Laden einer Regel aus dem Dateisystem, dem Repository oder der Zwischenablage.
  - **Speichern**

Zum Speichern der Regel im XML-Format in das Dateisystem, in das Repository oder in die Zwischenablage.

- **Drucken**
  Öffnet einen Drucken-Dialog, mit dem Sie die aktuell angezeigte Regel an Ihren Standarddrucker senden.
- **Von der Process Engine herunterladen…**
  Zum Anzeigen und Auswählen einer bereits publizierten oder mitgelieferten Regel.
  → Siehe *Regeln zur Wiederverwendung auf die inubit Process Engine publizieren (Workbench/Process Engine: Modul-Guide, Kap. 7.5, S. 162).*
- **Auf die Process Engine publizieren**
  (nur, wenn Sie den EDI-Regeleditor über das Menü „Extras“ öffnen)
  Lädt die aktuelle Regel als XML-Datei auf die inubit Process Engine. Die Regel steht damit zur Autoerkennung von Nachrichten zur Verfügung.
  → Siehe *Regeln zur Wiederverwendung auf die inubit Process Engine publizieren (Workbench/Process Engine: Modul-Guide, Kap. 7.5, S. 162).*
- **Regelstruktur validieren**
  (nur im Regel-Editor eines Adapters)
  Startet die Validierung der Regel gegen eine interne XML Schema-Datei.
- **Mapping-Template generieren…**
  Erzeugt eine Beispielnachricht mit allen Strukturinformationen der aktuell angezeigten Regel.
  Dieses Mapping-Template können Sie z. B. in einen XSLT Converter laden und zum Erzeugen einer Eingangsnachricht nutzen.
  → Siehe *XML-Eingangsnachricht für XML-EDI Adapter erstellen (Workbench/Process Engine: Modul-Guide, Kap. 7.3, S. 160).*
- **XML Schema generieren…**
  Erzeugt aus der Regel ein XML Schema, das Sie z. B. zum Validieren von Eingangsnachrichten nutzen können.
  → Für die Validierung können Sie einen XML Validator nutzen oder die Schemaprüfung aktivieren. Siehe *XML Validator (Utility) (Workbench/Process Engine: Modul-Guide, Kap. 26, S. 253)* und *Schemaprüfung der Modulnachrichten (Workbench: Benutzer-Guide, Kap. 2, S. 107)*.
- **Regeldarstellung**
  - **Attribut „id“:** ID, entspricht im Eigenschaftenbereich der Eingabe im Feld „ID“.

- **Attribut „xmlTag“:** XML-Elementname, entspricht im Eigenschaftenbereich der Eingabe im Feld „XML-Tag“.
- **Elementtyp:** Standardname des Regelknotens, z. B. Segment.
- **Beschreibung:** Zeigt im Baum neben jedem Element dessen Beschreibung an, entspricht im Eigenschaftenbereich der Eingabe im Feld „Beschreibung“.
- **Bedingte Option:** Zeigt im Baum an, ob ein Element optional (C für engl. „conditional“) oder erforderlich ist (M für engl. „mandatory“).
- **Wiederholung:** Zeigt im Baum die Häufigkeit des Vorkommens von Elementen an, z. B. [0…n].
- **Länge (oder Position des Elements):** Zeigt im Baum den Typ und die Länge bzw. Position von Elementen an, z. B. {an 1-7}.

- **Hilfsregel-Ansicht**
  Blendet einen zusätzlichen Bereich ein, in den eine weitere Regel als Vorlage für die neu zu erstellende Regel geladen werden kann.

- : Öffnet ein Eingabefenster für Zeichenketten, die im Regelbaum gesucht werden können.

Menü „Bearbeiten“

Regeldatei vor dem Publizieren überprüfen

Wenn aktiviert, dann wird jede Regel vor dem Publizieren gegen ein internes XML-Schema validiert, welches die Regelstruktur beschreibt. Wenn im *Dialog „Syntax und Validierung“ (Workbench/Process Engine: Modul-Guide, Kap. 7.8.3, S. 170)* die Option „XML-Validierung“ markiert ist, dann müssen alle Regeldateien dieser Struktur entsprechen, sonst kann das Modul nicht publiziert werden.

7.8 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Qualifierdefinition“, S. 167*
- *Dialog „EDI/XML-Adaptereigenschaften“, S. 168*
- *Dialog „Syntax und Validierung“, S. 170*
- *Dialog „Trennzeichen“, S. 172*
- *Dialog „XML/EDI-Adaptereigenschaften“, S. 173*

- *Assistent für neue Regel: Erstellungsmethode auswählen, S. 174*
- *Assistent für neue Regel: Neue Regel, S. 174*

7.8.1 Dialog „Qualifierdefinition“

Aufruf

Menü „Extras > EDI-Regel-Editor“, im EDI-Regel-Editor eine Regel öffnen, ein Element markieren > im Bereich „Element > Typ“ die Option „Qualifier“ markieren > Pfeil-Button anklicken.

In diesem Dialog können Sie einen Qualifier dem aktuell angezeigten Element zuordnen und neue Qualifier anlegen. Elemente mit dem Datentyp „Qualifier“ dürfen nur einen von mehreren Werten annehmen, die in einer benannten Liste definiert sind. Die Elementwerte können beim Einlesen der Nachrichten geprüft werden.

→ Siehe

- *Qualifier zu Element zuordnen (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 168)*
- *Qualifier hinzufügen und bearbeiten (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 168)*
- *Qualifierlisten prüfen (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 168)*

Dieser Dialog besteht aus folgenden Bereichen:

- **Links**: Liste aller bereits vorhandenen Qualifier.
- **Rechts**: Detailinformationen zu dem Qualifier, welcher in der Liste markiert ist.
  - **Qualifiername**
    Eindeutiger Name des Qualifiers.
  - **Beschreibung**
    Nähere Erläuterung des Qualifiers.
  - **Spalte „R“** („restricted“)
    Mit Hilfe dieser Spalte können Sie die Werte einschränken. Dies kann z. B. beim Erstellen von EDIFACT-Subsets nützlich sein. Es gilt:
    - Wenn kein Wert markiert ist, dürfen alle Werte verwendet werden.
    - Sobald ein Wert markiert ist, dürfen nur noch die markierten Werte verwendet werden.

Dynamische Werte können nicht genutzt werden.

Die Werte müssen eindeutig sein. Alle Werte in einer Liste müssen

eineindeutig zu allen anderen Werten in der Liste sein. Dies gilt insbesondere, wenn das Element, das einen Qualifier als Typen benutzt, den Identifier für ein Segment oder eine Segmentgruppe darstellt.
Sie dürfen daher z. B. nicht folgende Werte verwenden:
`B`
`BB`
`BBB`
Denn `B` ist enthalten in `BB` und `BBB`, `BB` ist enthalten in `BBB`. Wenn der EDI-Parser ein Element findet, dass mit `BBB` beginnt, dann ist nicht klar, wie dieses interpretiert werden soll: `B | B | B` oder `B | BB` oder `BBB` oder `BB | B` usw.

Funktionen im Dialog

- **Qualifier zu Element zuordnen**
  Um dem aktuell ausgewählten Element einen Qualifier zuzuordnen, markieren Sie den Qualifier in der Liste und klicken auf „OK“.
- **Qualifier hinzufügen und bearbeiten**
  Nutzen Sie die Werkzeugleiste über der Tabelle.
- **Qualifierlisten prüfen**
  Um Elemente mit dem Datentyp „Qualifier“ beim Einlesen der Nachrichten prüfen zu lassen, markieren Sie in den Moduleigenschaften die Option „Feldwerte gegen Qualifierliste prüfen“. Wenn die Werte in den Nachrichten nicht der vorgegebenen Qualifierliste entsprechen, dann wird dies als Fehler gewertet.
  → Siehe *Feldwerte gegen Qualifierliste prüfen (Workbench/ Process Engine: Modul-Guide, Kap. 7, S. 171)*.

7.8.2 Dialog „EDI/XML-Adaptereigenschaften“

Autoerkennung der Regeldatei

- **Autoerkennung für EDIFACT, X12 und HL7**
  Wenn markiert, dann wird versucht, das Format der eingehenden Nachrichten automatisch zu erkennen und eine passende Konvertierungsregel zuzuordnen. Wenn keine passende Regeldatei vorhanden ist, wird ein Fehler gemeldet.
  Eine Liste aller EDIGAS-Nachrichtenarten, die der EDI Adapter zur Autoerkennung nutzt, finden Sie auf der inubit Process Engine unter `<iS-installdir>/server/ibis_root/conf/edi/EDIGAS-MESSAGES.xml`. Damit neue EDIGAS-Nachrichtenarten automatisch erkannt werden, müssen Sie diese in die Datei `EDIGAS-MESSAGES.xml` mit aufnehmen. Erweitern Sie dazu die

XML-Struktur um ein neues Element `message` mit dem Attribut `name` und dem entsprechenden Wert der neuen EDIGAS-Nachrichtart. Der EDI Adapter erkennt nun diese Nachrichtenart automatisch.

- **Erweiterte Autoerkennung**
  Diese Option sorgt dafür, dass EDIFACT-Unterversionen auf Basis desselben Jahrgangs und derselben Version erkannt werden.
  Im Normalfall nutzt der EDI Adapter die ersten drei Elemente aus dem UNH-Segment (Jahrgang und Version) zum Auffinden der passenden Regeldatei. Wenn diese Option markiert ist, dann nutzt der EDI Adapter alle Elemente.

Namen der XML-Elemente

- **Attribut „id“ verwenden**
  Wenn markiert, enthält die XML-Nachricht Elemente mit abgekürzten Bezeichnern (`S_<ID>` bzw. `D_<ID>`).
- **Attribut „xmlTag“ verwenden**
  Wenn markiert, dann wird die Nachricht mit sprechenden Elementnamen ausgegeben, entsprechend dem Wert für „XML Tag“ in der Konvertierungsregel.
  „Sprechende“ Elementnamen verbessern die Lesbarkeit der Nachricht, erhöhen aber die Dateigröße und beeinflussen damit die Performance.

Formatierung der EDI-Nachricht

- **Zeilenendezeichen/Zeilenlänge**
  Wenn die EDI-Dateien für die Anzeige in einem Betriebssystem aufbereitet werden sollen, dass bestimmte Zeilenendezeichen oder eine maximale Zeilenlänge voraussetzt, müssen Sie diese Daten angeben, damit die Nachrichten geeignet aufbereitet werden können.
  Beispiel: EDIFACT-Nachrichten haben üblicherweise keinen Zeilenumbruch, während das Betriebssystem OS/400 Zeilen bis max. 80 Zeichen erlaubt. Hier müssten Sie als Zeilenendezeichen „\n“ eingeben und als Zeilenlänge „80“.

Das Zeilenendezeichen darf nicht identisch sein mit den Trennzeichen in der Nachricht!

- **Zeichensatzkodierung**
  Zeichensatz der Eingangsnachrichten. Mit der Angabe stellen Sie sicher, dass die Eingangsnachricht korrekt konvertiert wird und z. B. Umlaute richtig dargestellt werden.

Standardmäßig ist der Zeichensatz ISO-8859-1 ausgewählt, der alle Zeichen der gebräuchlichen westeuropäischen Sprachen und deren Sonderzeichen enthält.

Formatierung der XML-Nachricht

- **Keine Einrückungen/Zeilenumbrüche hinter Elementen, alle Whitespaces werden entfernt**
  Der Name der Option ist sprechend.
- **Entferne führende Nullen, die zum Auffüllen von numerischen Feldern eingefügt wurden**
  Der Name der Option ist sprechend.
- **EDIFACT-Nachrichten-Gruppe zur XML-Nachricht hinzufügen**
  Die Ausgangsnachricht enthält zusätzlich den Body der EDI-Eingangsnachricht.
- **Dezimalfelder mit einheitlichem Trennzeichen ausgeben**
  Wenn markiert, wird das Trennzeichen für alle Dezimalfelder einheitlich gesetzt. Nutzen Sie die Auswahlliste, um das Dezimaltrennzeichen zu spezifizieren:
  - „.“ (Punkt): In allen Dezimalfeldern wird ein Punkt zum Abtrennen der Nachkommastellen verwendet.
  - „,“ (Komma): In allen Dezimalfeldern wird ein Komma zum Abtrennen der Nachkommastellen verwendet.
  - „UNA spezifisch“: Es wird das Dezimaltrennzeichen aus dem UNA-Segment zum Abtrennen der Nachkommastellen verwendet. Wenn „UNA spezifisch“ ausgewählt ist, aber kein UNA-Segment vorhanden ist, dann wird standardmäßig ein Punkt als Trennzeichen verwendet.

7.8.3 Dialog „Syntax und Validierung“

Um eine EDI-Nachricht vollständig gegen die Regeldatei zu prüfen, aktivieren Sie die Optionen „XML-Validierung“, „Syntaxfehler in EDI-Eingangsnachricht ignorieren“ und „Feldwerte gegen Qualifierliste prüfen“.

Analyse

- **Regeldatei cachen**
  Wenn markiert, dann bleibt die Regeldatei während der Ausführung des gesamten Workflows im Speicher.
  Diese Einstellung ist sinnvoll, um viele Nachrichten hintereinander zu konvertieren, weil der Konvertierungsprozess schneller ist, wenn die Regeldatei nicht immer wieder neu geladen werden muss. Caching ist sinnvoll bei Nachrichten ab einer Größe von 300 KByte.

Berücksichtigen Sie bei der Dimensionierung des Arbeitsspeichers, dass das Cachen zusätzlich Arbeitsspeicher kostet.

- **Syntaxfehler in EDI-Eingangsnachricht ignorieren**
  Wenn Sie diese Option auswählen, werden fehlerhaft formatierte Bereiche in der Nachricht übergangen und die Konvertierung wird trotz der Fehler fortgesetzt.

Wenn Bereiche in einer Eingangsnachricht fehlerhaft formatiert sind und übergangen werden, dann fehlen diese in der Ausgangsnachricht! Um fehlerhafte Bereiche zu identifizieren, aktivieren Sie das Server Trace Log. Danach werden die Fehlermeldungen, die bei der Benutzung des EDI Format Adapters entstehen, protokolliert. Siehe *Server Trace Log aktivieren und konfigurieren (Process Engine: Administrator- und Entwickler-Guide, Kap. 2.1.3, S. 23).*

- **Fehlerinformationen zur Ausgangsnachricht hinzufügen**
  Aktivieren Sie diese Option, um die übergangenen und daher fehlenden Bereiche zu identifizieren.

Aktivieren Sie diese Option **nicht** gleichzeitig mit der XML-Validierung. Sonst erzeugen die zusätzlichen Fehlerinformationen weitere Fehler, wenn die Ausgangsnachricht der XML Validierung übergeben wird.

- **EDIFACT CONTRL-Nachricht erzeugen**
  Generiert automatisch CONTRL-Nachrichten. Abhängig vom Verarbeitungsergebnis wird eine Error- oder eine Acknowledge-CONTRL-Nachricht erzeugt und in eine der folgenden Variablen geschrieben:
  - Error: `edifact.contrl.reject`
  - Acknowledged: `edifact.contrl.acknowledged`

Validierung

- **Feldwerte gegen Qualifierliste prüfen**
  Wenn markiert, dann werden Elementwerte vom Typ „Qualifier“ beim Einlesen der Eingangsnachricht geprüft.
  → Siehe *Dialog „Qualifierdefinition“ (Workbench/Process Engine: Modul-Guide, Kap. 7.8.1, S. 167).*
- **XML-Validierung**
  Wenn aktiviert, dann wird die XML-Nachricht gegen ein XML Schema validiert, das nach jeder Regeländerung automatisch aus der Konvertierungsregel erzeugt wird. Bei der Validierung werden auch der Elementtyp und die Qualifierlisten überprüft, anders als bei der standardmäßig immer durchgeführten automatischen Syntaxprüfung.

Aktivieren Sie die Option „Syntaxfehler ignorieren", wenn Sie die XML-Validierung nutzen. Dann werden alle Prüfungen gegen das XML Schema durchgeführt und im Fehlerfall wird nur ein Fehlertyp ausgegeben. Wenn die Option „Syntaxfehler ignorieren" deaktiviert ist, dann erzeugt eine fehlerhafte Nachricht mit einem zu langen Element einen anderen Fehlertyp als eine Nachricht mit einem falschen Qualifier.
Bei einer großen Anzahl von Nachrichten reduziert diese Option die Performance.

- **Bei fehlerhafter Schema-Validierung Fehlerausgang verwenden**
  Wenn diese Option aktiviert ist, dann wird die Ausführung des Workflows abgebrochen, sobald die Schema-Validierung fehlschlägt. Die Nachricht wird an den Fehlerausgang weitergeleitet.
- **Schemastil**

Aktivieren Sie diese Option nur, wenn Sie Probleme bei der Validierung Ihrer Nachricht gegen das aus der Regel erzeugte Schema feststellen.

- **Russian Doll Design** (Elemente sind inline definiert)
  Standarddesign für Nicht-EDIFACT-Regeldateien.
  Die Schema-Struktur spiegelt die Instanz-Struktur wider: Alle Elementdeklarationen sind ineinander geschachtelt, es gibt keine Referenzen auf identische Elemente.
- **Venetian Blind Design** (Elemente sind referenziert)
  Standarddesign für EDIFACT-Regeldateien.
  Die Deklarationen der Elementtypen werden anhand des Elementnamens global abgelegt. Die Elemente verweisen auf die entsprechende Deklaration.

7.8.4 Dialog „Trennzeichen“

Delimitererkennung

- **Autoerkennung für EDIFACT, X12 und HL7**
  Bei EDIFACT, ANSI X12 und HL7 können Sie die Autoerkennung der Trennzeichen aktivieren. Für alle anderen Standards müssen Sie die Trennzeichen manuell angeben.

Auch die Autoerkennung kann nicht immer alle Trennzeichen korrekt erkennen. Prüfen Sie daher das Ergebnis!

Standard

Wählen Sie einen Standard aus, um die Felder im Bereich „Trennzeichen“ vorzubelegen.

Es werden immer nur die Standards angezeigt, für deren Regeln Sie eine Lizenz erworben haben!

Trennzeichen

Wenn Sie einen Standard ausgewählt haben, dann sind die Felder bereits korrekt vorbelegt.

Sie können die vorgegebenen Werte überschreiben, indem Sie andere Werte aus den Listen auswählen oder Werte manuell eingeben.

EDIFACT

(nur bei XML/EDI Adaptern)

- **Erzeuge EDIFACT UNA-Segment**
  Fügt als erstes Segment eine Zeichenkette in die EDI-Nachricht ein, welche die verwendeten Trennzeichen beschreibt, z. B. UNA:+.? '.
  Diese Option ist nur bei der Verwendung von EDIFACT-Regeldateien sinnvoll.

7.8.5 Dialog „XML/EDI-Adaptereigenschaften“

Bis auf die folgenden Optionen ist dieser Dialog identisch mit dem *Dialog „EDI/XML-Adaptereigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 7.8.2, S. 168)*.

Formatierung der XML-Nachricht

- **Optionale Elemente der Regeldatei in die Ausgangsnachricht einfügen**
  - Option ist markiert:
    Wenn die Regeldatei optionale Elemente enthält, die in der eingehenden XML-Nachricht nicht mit Werten gefüllt sind, dann werden in der ausgehenden EDI-Nachricht für diese optionalen Elemente leere EDI-Strukturen erzeugt.
  - Option ist nicht markiert:
    Die entsprechenden Elemente werden ignoriert.
- **Leere XML-Tags und -Strukturen ignorieren**
  Wenn die Option markiert ist, dann werden Elemente ohne Wert aus der eingehenden XML-Nachricht nicht nach EDI konvertiert.

Diese Option wird nicht auf Nachrichten mit fest positionierten Elementen angewendet, siehe *Hierarchische Struktur mit fest positionierten Elementen (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 175)*.

Leere Elemente können z. B. bei einer früheren Konvertierung von EDI nach XML entstanden sein, als für optionale Bestandteile der XML-Datei leere XML-Strukturen angelegt wurden, damit die Strukturprüfung erfolgreich durchgeführt werden konnte.

7.8.6 Assistent für neue Regel: Erstellungsmethode auswählen

→ Siehe *EDI-XML Adapter erstellen und Regel konfigurieren (Workbench/Process Engine: Modul-Guide, Kap. 7.2, S. 158).*

Die Nutzung der EDI-Regeln ist lizenzpflichtig: Es werden immer nur die EDI-Standards angezeigt, für deren Regeln Sie eine Lizenz erworben haben!

7.8.7 Assistent für neue Regel: Neue Regel

Nachrichtenstandard

- **Standard**
  Vorauswahl für die zu erstellende Regel. Wenn der gewünschte Standard nicht angeboten wird, wählen Sie „Custom“, um eine vollständig neue Regel zu erstellen.

Aus dem SWIFT-Standard werden aktuell nur MT940-Nachrichten unterstützt und verarbeitet, die zur elektronischen Übermittlung von Kontoauszugsdaten verwendet werden. Momentan ist nur die Konvertierung von MT940-Nachrichten nach XML möglich, nicht umgekehrt.

Die Nutzung der EDI-Regeln ist lizenzpflichtig: Es werden immer nur die EDI-Standards angezeigt, für deren Regeln Sie eine Lizenz erworben haben!

- **Eigener Name**

Aktiv, wenn als Standard „Custom“ gewählt ist. Ist der Name des neu zu erzeugenden Standards.

- **Struktur**
  Aktiv, wenn als Standard „Custom“ angegeben ist. Wählen Sie eine Regelstruktur aus, im Regeleditor werden entsprechend dieser Struktur-Vorauswahl die passenden Bedienelemente zum komfortablen Aufbau Ihrer Regel eingeblendet:
  a. **Hierarchische Struktur mit Elementbegrenzung**
     z. B. EDIFACT
     - Hierarchische Struktur:
       Eine Nachricht kann mehrere Messages enthalten, eine Message kann aus Gruppen > Segmenten > Elementen bestehen
     - Elemente haben variable Längen, sind getrennt durch Trennzeichen

  b. **Hierarchische Struktur mit fest positionierten Elementen**
     z. B. VDA
     Wie a., jedoch mit festen Längen, nicht-belegte Stellen werden mit Füllzeichen aufgefüllt

  c. **Gemischte Segmente mit Elementbegrenzung**
     z. B. DATANORM
     Wie a., jedoch ohne hierarchische Struktur

Eigenschaften

Geben Sie einen Namen für die zu erstellende Regel an. Alle anderen Angaben sind optional und können auch nachträglich geändert werden.

→ Siehe *Neue Regel erstellen (Workbench/Process Engine: Modul-Guide, Kap. 7.4, S. 161).*

Dieser Abschnitt erläutert die folgenden Themen:

- *Funktionsprinzip, S. 177*
- *Regel-Editor (Oberfläche), S. 180*
- *Neue Regel erstellen, S. 182*
- *Dialogbeschreibungen, S. 183*

Verwendung

Flatfiles sind Dateien, die Datensätze ohne explizite Informationen über ihre Strukturen enthalten. Mit Hilfe eines Flat Adapters werden Flatfiles in XML-Dateien bzw. XML-Dateien in Flatfiles konvertiert.

Für die Konvertierung werden XML-basierte Regeln verwendet. Für jeden Nachrichtentyp muss eine spezifische Regel erstellt werden, die dann für beide Konvertierungsrichtungen verwendet werden kann. In der Regel wird festgelegt, wie die Flatfile-Elemente auf Flat-XML-Elemente abgebildet werden.

→ Beachten Sie auch die Dokumentation zum *EDI Adapter (Format Adapter) (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 155)*.

Wenn Sie Flatfile-Eingangsnachrichten nach XML konvertieren, die das Zeichen 0x00 enthalten, dann werden Ausgangsnachrichten mit einer ungültigen Entity-Referenz erzeugt!
Ersetzen Sie das Zeichen 0x00 vor der Konvertierung mit einem Data Stream Modifier durch ein anderes Zeichen, das in Ihren Eingangsnachrichten nicht vorkommt. Siehe *Data Stream Modifier (Utility) (Workbench/Process Engine: Modul-Guide, Kap. 15, S. 213).*

8.1 Funktionsprinzip

Ein Flatfile hat eine nicht-hierarchische Struktur aus einem oder mehreren Segmenten, die jeweils ein oder mehrere Elemente enthalten. Implizite Segment- und Elementeigenschaften wie z. B. die Häufigkeit eines Elements in einem Segment werden im XML entweder

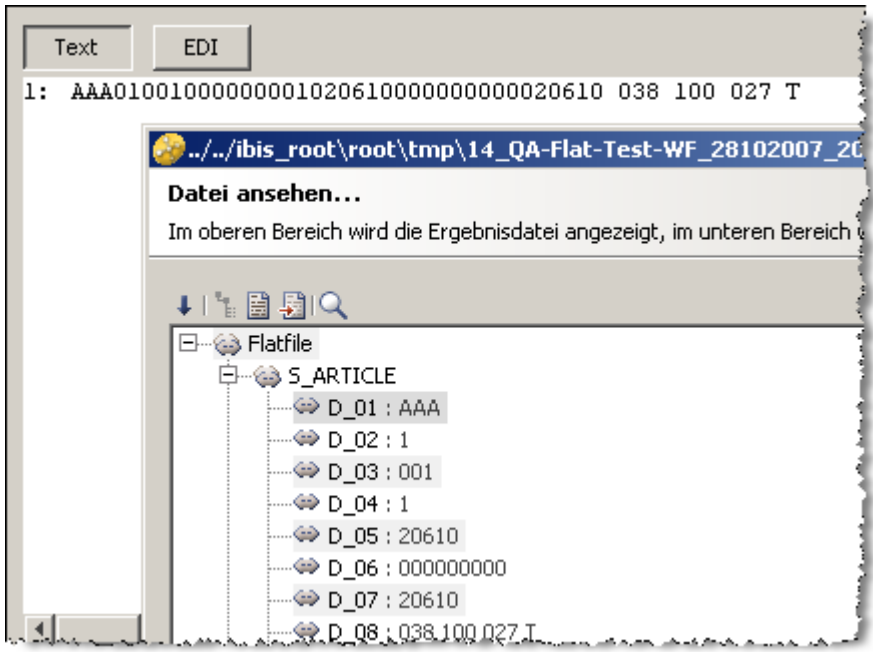
- explizit als Attributwert der betreffenden XML-Elemente notiert, z. B. loop=3
- oder beim Anlegen des Flat Adapter als Adapter-Eigenschaft konfiguriert, wie z. B. das Trennzeichen, welches das Segmentende signalisiert.

Die folgende Tabelle zeigt, wie Segmente und Elemente auf XML-Elemente abgebildet werden.

Hierarchiestufe	Struktur im Flatfile	Struktur im Flat-XML
1	Kein root-element	root-Element `<Flatfile>` Name nicht änderbar!
2	Segmente	XML-Element `<Segment>`
	Eigenschaften	Attribute
	kein Segmentname	ID und xmlTag, Werte beliebig
	ein oder mehrere Segmente Häufigkeit des Vorkommens ist festgelegt	loop=n|[Anzahl]
	Pflichtsegment oder optional	required=C(onditional)|M(andatory)
	jedes Segment kann ein oder mehrere Elemente enthalten	kann über ein XML Schema für das Template festgelegt werden
	Elemente sind durch Trennzeichen getrennt oder ohne Unterbrechung aufeinander folgend	Angabe beim Konfigurieren des Adapters
3	Elemente	XML-Element <Element>
	Eigenschaften	Eigenschaften analog zu Segment
	kein Elementname	ID und name, Wert beliebig
	festgelegte Länge	Längenangabe über Anfangs- und Endposition
	festgelegte Position Beispiel: Die Angabe {n 15-23} bedeutet, dass es sich um einen numerischen Wert handelt, der in Position 15 beginnt und in Position 23 endet.	z.B. posStart=1, posEnd=14
	nicht vorhandene Zeichen	
	festgelegter Datentyp	type=(AN|N)
	Pflichtelement oder optionales Element	required=C(onditional)|M(andatory)

Beispiel

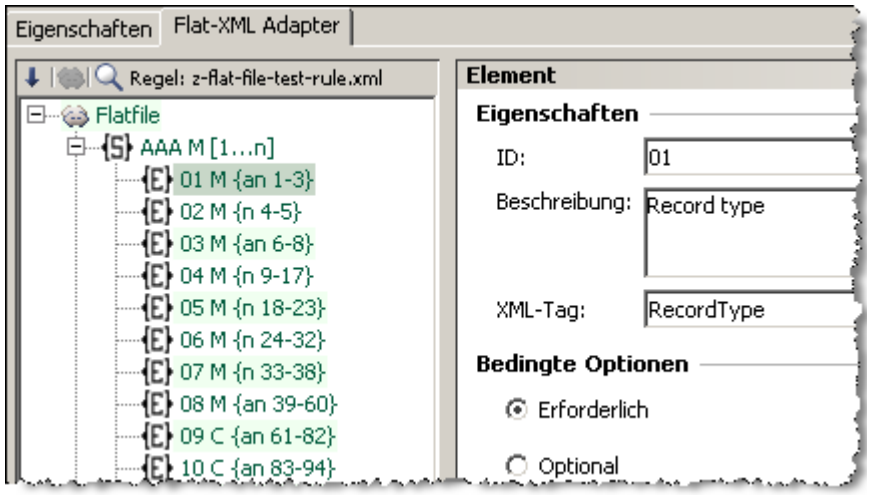
Die folgende Abbildung zeigt ein Flatfile (im Hintergrund) und die daraus erstellte Flat-XML-Nachricht (Vordergrund):



Das Flatfile (im Hintergrund) besteht aus mehreren Segmenten, die durch einen Zeilenumbruch (nicht sichtbar) getrennt sind.

In der Flat-XML-Nachricht wurde das Root-Element `<Flatfile>` hinzugefügt. Für jedes Flatfile-Element wurde ein XML-Element erzeugt und entsprechend der Konvention „D_n+1“ (D_01, D_02…) benannt. Bei den Elementwerten wurden führende Nullen entfernt: aus dem zweiten Element „01“ im Flatfile wird im Flat-XML-File das Element D_02 mit dem Wert „1“.

Die folgende Abbildung zeigt einen Ausschnitt aus einer Regel, die zum Konvertieren der oben abgebildeten Dateien verwendet wurde:



Dieser Ausschnitt zeigt, dass

- obligatorisch das root-Element `<Flatfile>` eingefügt worden ist
- das erste Segment vorhanden sein muss („M“ in der Baumdarstellung)

- es unbegrenzt oft vorkommen kann ([1…n] in der Baumdarstellung)
- es aus einer Reihe von Elementen besteht, u. a. aus dem Element 01, das ebenfalls obligatorisch ist (M), einen alphanumerischen Datentyp hat, in Position 1 beginnt und in Position 3 endet ({an 1-3} in der Baumdarstellung).

8.2 Regel-Editor (Oberfläche)

Im Flat Rule Editor erstellen Sie Regeln.

→ Beachten Sie auch die Dokumentation zum *EDI Adapter (Format Adapter) (Workbench/Process Engine: Modul-Guide, Kap. 7, S. 155)*.

Der Editor besteht aus folgenden Bereichen:

1. **Hilfsregelbereich**

   Initial nicht sichtbar. Einzublenden über den -Button > „Hilfsregel-Ansicht.

   In diesen Bereich können Sie eine zusätzliche Regel laden und als Vorlage für Ihre neue Regel verwenden. Sie können per Drag'n'Drop Elemente und Segmente aus der Hilfsregel in den Regel-Bereich ziehen und damit eine neue Regel erstellen.
2. **Regelbereich**

   Zum Erstellen einer neuen Regel.

   Zum Einfügen, Verschieben und Löschen von Segmenten/ Elementen nutzten Sie das Kontextmenü des Baums.

   Um die Darstellung des Baums zu ändern, wählen Sie -Button > „Regeldarstellung".
3. **Eigenschaftenbereich**

   Zeigt die Eigenschaften des Elements/Segments an, welches in der Regel markiert ist.

Menüleiste

In der Menüleiste des Editors gibt es folgende Befehle:

- : Öffnet ein Eingabefenster für Zeichenketten, die im Regelbaum gesucht werden können.
- Öffnet ein Menü mit folgenden Befehlen:
  - **Neu**: Erzeugt eine neue Regeldatei mit dem root-Element `<Flatfile>`.

- **Öffnen**: Zum Anzeigen einer im Dateisystem gespeicherten Regel.
- **Speichern**: Speichert die aktuell angezeigte Regel.
- **Drucken**
  Öffnet einen Drucken-Dialog, mit dem Sie die aktuell angezeigte Regel an Ihren Standarddrucker senden.
- **Regelstruktur validieren**
  Startet die Validierung der Regel gegen eine interne XML Schema-Datei.
- **Mapping-Template generieren**
  Öffnet den Assistenten zum Erzeugen einer Flat-XML-Vorlage, die alle Strukturinformationen der aktuell angezeigten Regel enthält. Diese Vorlage kann im XSLT-Converter als Zieldatei geladen werden.

Alternativ verwenden Sie im XSLT-Converter den Regeldatei-Explorer, der das Erzeugen, Speichern und Laden der Regel für Sie übernimmt.

- **XML-Schema generieren**
  Erzeugt aus der Regel ein XML-Schema, das Sie z. B. zum Validieren von Eingangsnachrichten nutzen können.
  → Für die Validierung können Sie einen XML Validator nutzen oder die Schemaprüfung aktivieren. Siehe *XML Validator (Utility) (Workbench/Process Engine: Modul-Guide, Kap. 26, S. 253)* und *Schemaprüfung der Modulnachrichten (Workbench: Benutzer-Guide, Kap. 2, S. 107)*.
- **Excel-Report generieren**
  Öffnet einen Assistenten, in dem Sie detailliert festlegen, welche Informationen aus der Regel in den Excel-Report exportiert werden sollen. Neben dem Export ins xls-Format, können Sie auch eine Option wählen, mit der die Regeldateien nach CSV exportiert werden.

- **Regeldarstellung**
  Zum Ändern der Regel-Ansicht:
  - **Attribut „id“:** ID, entspricht im Eigenschaftenbereich der Eingabe im Feld „ID“.
  - **Attribut „xmlTag“:** XML-Elementname, entspricht im Eigenschaftenbereich der Eingabe im Feld „XML-Tag“.
  - **Elementtyp:** Standardname des Regelknotens, z. B. Segment.
  - **Beschreibung**
    Zeigt im Baum neben jedem Element dessen Beschreibung an, entspricht im Eigenschaftenbereich der Eingabe im Feld „Beschreibung“.
  - **Bedingte Option**

Zeigt im Baum an, ob ein Element optional (C für engl. „conditional“) oder erforderlich ist (M für engl. „mandatory“).

- **Wiederholung**
  Zeigt im Baum die Häufigkeit des Vorkommens von Elementen an, z. B. [0…n].
- **Länge (oder Position des Elements)**
  Zeigt im Baum den Typ und die Länge bzw. Position von Elementen an, z. B. {an 1-7}.

- **Hilfsregel-Ansicht**
  Blendet einen zusätzlichen Bereich ein, in den eine weitere Regel als Vorlage für die neu zu erstellende Regel geladen werden kann.

Menü „Bearbeiten“

Regeldatei vor dem Publizieren überprüfen

Wenn aktiviert, dann wird jede Regel vor dem Publizieren gegen ein internes XML-Schema validiert, welches die Regelstruktur beschreibt. Wenn im *Dialog „Syntax und Validierung“ (Workbench/Process Engine: Modul-Guide, Kap. 7.8.3, S. 170)* die Option „XML-Validierung“ markiert ist, dann müssen alle Regeldateien dieser Struktur entsprechen, sonst kann das Modul nicht publiziert werden.

8.3 Neue Regel erstellen

Zum Erstellen einer Konvertierungsregel benötigen Sie detaillierte Informationen über die Struktur der Nachrichten, die Sie mit dem Adapter verarbeiten möchten!

So gehen Sie vor

1. Hilfsregel laden (wenn nötig)
   Hilfsregeln sind normale Regeln, die Sie zuvor erstellt haben oder die Ihnen zur Verfügung gestellt wurden. Standardmäßig werden mit der inubit Suite 6 keine Regeln ausgeliefert.
   a. Klicken Sie im Regelbereich auf den ⬇-Button und wählen Sie „Hilfsregel-Ansicht“. Der Hilfsregel-Bereich wird links vom Regel-Bereich eingeblendet.
   b. Klicken Sie in dem leeren Bereich auf den ⬇-Button und wählen Sie „Öffnen“. Sie können Dateien aus dem Dateisystem, dem Repository und aus der Zwischenablage laden.

c. Navigieren Sie zu der Regel, die Sie als Hilfsregel anzeigen möchten und wählen Sie diese aus. Die Regel wird angezeigt.

2. **Neue Regel erzeugen**

   Klicken Sie im Regelbereich auf den -Button und wählen Sie „Neu > Regel…“. Eine neue Regel mit dem Root-Element `Flatfile` wird angezeigt.

3. **Neue Regel bearbeiten**

   Sie haben folgende Möglichkeiten:

   - Im Regelbereich mit Hilfe des Kontextmenüs Segmente und Elemente hinzufügen
   - Per Drag’n’Drop Segmente und Elemente aus der Hilfsregel verwenden

Die Regel wird automatisch zusammen mit dem Adapter gespeichert.

8.4 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Flatfile-XML Adaptereigenschaften“, S. 183*
- *Dialog „XML-Flatfile Adapter Eigenschaften“, S. 185*

8.4.1 Dialog „Flatfile-XML Adaptereigenschaften“

Name der XML-Elemente

- **Attribut „id“ verwenden**

  Wenn markiert, enthält die XML-Nachricht Elemente mit abgekürzten Bezeichnern (`S_<ID>` bzw. `D_<ID>`).

- **Attribut „xmlTag“ verwenden**

  Wenn markiert, dann wird die Nachricht mit sprechenden Elementnamen ausgegeben, entsprechend dem Wert für „XML Tag“ in der Konvertierungsregel.

  „Sprechende“ Elementnamen verbessern die Lesbarkeit der Nachricht, erhöhen aber die Dateigröße und beeinflussen damit die Performance.

Formatierung der Flat-Nachricht

- **Segmenttrennzeichen**
  Wählen Sie einen Wert aus der Liste aus, um anzugeben, wie die Segmente in den Eingangsnachrichten voneinander getrennt sind.
- **Dezimaltrennzeichen (Eingabe)**
  Geben Sie an, durch welches Zeichen (Punkt oder Komma) Dezimalzahlen mit Nachkommastellen getrennt sind.
- **Dezimaltrennzeichen (Ausgabe)**
  Das Dezimaltrennzeichen kann bei der Konvertierung ersetzen werden. Legen Sie fest, welches Zeichen verwendet werden soll (Punkt oder Komma).
- **Zeichensatzkodierung**
  Zeichensatz der Eingangsnachrichten. Mit der Angabe stellen Sie sicher, dass die Eingangsnachricht korrekt konvertiert wird und z. B. Umlaute richtig dargestellt werden.
  Standardmäßig ist der Zeichensatz ISO-8859-1 ausgewählt, der alle Zeichen der gebräuchlichen westeuropäischen Sprachen und deren Sonderzeichen enthält.

Formatierung der XML-Nachricht

- **Entferne führende Nullen, die zum Auffüllen von numerischen Feldern eingefügt wurden**
  Der Name der Option ist sprechend.
- **Leere Elemente nicht erzeugen**
  Wenn diese Option markiert ist, dann werden für Flatfile Elemente, die nur Whitespaces enthalten, keine XML-Elemente erzeugt. Dadurch werden die Ausgangsnachrichten kleiner und XSLT-Stylesheets für die evtl. anschließende Verarbeitung sind besser zu pflegen.
- **Keine Einrückungen/Zeilenumbrüche hinter Elementen, alle Whitespaces werden entfernt**
  Der Name der Option ist sprechend.

Analyse

- **Regeldatei cachen**
  Wenn markiert, dann bleibt die Regeldatei während der Ausführung des gesamten Workflows im Speicher.
  Diese Einstellung ist sinnvoll, um viele Nachrichten hintereinander zu konvertieren, weil der Konvertierungsprozess schneller ist, wenn die Regeldatei nicht immer wieder neu geladen werden muss. Caching ist sinnvoll bei Nachrichten ab einer Größe von 300 KByte.

Berücksichtigen Sie bei der Dimensionierung des Arbeitsspeichers, dass das Cachen zusätzlich Arbeitsspeicher kostet.

- **Syntaxfehler in der Eingangsnachricht ignorieren**

Wenn Sie diese Option auswählen, wird die Konvertierung im Fehlerfall nicht abgebrochen. Stattdessen werden fehlerhafte Bereiche in der Nachricht übergangen.

Wenn die Option deaktiviert ist, dann wird im Fehlerfall eine Fehlermeldung angezeigt und die Verarbeitung abgebrochen. Deaktivieren Sie diese Option zum Testen einer Regel, damit Sie eventuelle Fehler darin finden können.

- **Fehlerinformationen zur Ausgabe XML-Nachricht hinzufügen**

  Fügt die Originalsegment- und Elementdaten in die Flat XML Ausgangsnachrichten ein.

Validierung

- **XML-Validierung**

  Wenn aktiviert, dann werden die Ausgangsnachrichten gegen ein XML Schema validiert, das aus der Konvertierungsregel erzeugt wurde. Bei der Validierung werden auch der Elementtyp und die Qualifierlisten überprüft, anders als bei der standardmäßig immer durchgeführten automatischen Syntaxprüfung.

Aktivieren Sie die Option „Syntaxfehler in der Eingangsnachricht ignorieren", wenn Sie die XML Validierung nutzen. Dann werden alle Prüfungen gegen das XML Schema durchgeführt und im Fehlerfall wird nur ein Fehlertyp ausgegeben. Wenn die Option deaktiviert ist, dann erzeugt eine fehlerhafte Nachricht mit einem zu langem Element einen anderen Fehlertyp als eine Nachricht mit einem falschen Qualifier.

- **Bei fehlerhafter Schema-Validierung Fehlerausgang verwenden**

  Wenn diese Option aktiviert ist, dann wird die Ausführung des Workflows abgebrochen, sobald die Schema-Validierung fehlschlägt. Die Nachricht wird an den Fehlerausgang weitergeleitet.

8.4.2 Dialog „XML-Flatfile Adapter Eigenschaften“

Name der XML-Elemente

- **Attribut „id“ verwenden**

  Wenn markiert, enthält die XML-Nachricht Elemente mit abgekürzten Bezeichnern (`S_<ID>` bzw. `D_<ID>`).

- **Attribut „xmlTag“ verwenden**

Wenn markiert, dann wird die Nachricht mit sprechenden Elementnamen ausgegeben, entsprechend dem Wert für „XML Tag“ in der Konvertierungsregel.

„Sprechende“ Elementnamen verbessern die Lesbarkeit der Nachricht, erhöhen aber die Dateigröße und beeinflussen damit die Performance.

Formatierung der Flat-Nachricht

- **Segmenttrennzeichen (Eingabe)**
  Wählen Sie einen Wert aus der Liste aus, um anzugeben, wie die Segmente in den Eingangsnachrichten voneinander getrennt sind.
- **Dezimaltrennzeichen (Eingabe)**
  Geben Sie an, durch welches Zeichen (Punkt oder Komma) Dezimalzahlen mit Nachkommastellen getrennt sind.
- **Dezimaltrennzeichen (Ausgabe)**
  Das Dezimaltrennzeichen kann bei der Konvertierung ersetzen werden. Legen Sie fest, welches Zeichen verwendet werden soll (Punkt oder Komma).
- **Zeichensatzkodierung**
  Zeichensatz der Eingangsnachrichten. Mit der Angabe stellen Sie sicher, dass die Eingangsnachricht korrekt konvertiert wird und z. B. Umlaute richtig dargestellt werden.
  Standardmäßig ist der Zeichensatz ISO-8859-1 ausgewählt, der alle Zeichen der gebräuchlichen westeuropäischen Sprachen und deren Sonderzeichen enthält.

Analyse

- **Regeldatei cachen**
  Wenn markiert, dann bleibt die Regeldatei während der Ausführung des gesamten Workflows im Speicher.
  Diese Einstellung ist sinnvoll, um viele Nachrichten hintereinander zu konvertieren, weil der Konvertierungsprozess schneller ist, wenn die Regeldatei nicht immer wieder neu geladen werden muss. Caching ist sinnvoll bei Nachrichten ab einer Größe von 300 KByte.

Berücksichtigen Sie bei der Dimensionierung des Arbeitsspeichers, dass das Cachen zusätzlich Arbeitsspeicher kostet.

-
- **Syntaxfehler in der Eingangsnachricht ignorieren**
  Wenn Sie diese Option auswählen, wird die Konvertierung im Fehlerfall nicht abgebrochen. Stattdessen werden fehlerhafte Bereiche in der Nachricht übergangen.

Wenn die Option deaktiviert ist, dann wird im Fehlerfall eine Fehlermeldung angezeigt und die Verarbeitung abgebrochen. Deaktivieren Sie diese Option zum Testen einer Regel, damit Sie eventuelle Fehler darin finden können.

Validierung

- **XML-Validierung**
  Wenn aktiviert, dann werden die Ausgangsnachrichten gegen ein XML Schema validiert, das aus der Konvertierungsregel erzeugt wurde. Bei der Validierung werden auch der Elementtyp und die Qualifierlisten überprüft, anders als bei der standardmäßig immer durchgeführten automatischen Syntaxprüfung.
  - **Bei fehlerhafter Schema-Validierung Fehlerausgang verwenden**
    Wenn diese Option aktiviert ist, dann wird die Ausführung des Workflows abgebrochen, sobald die Schema-Validierung fehlschlägt. Die Nachricht wird an den Fehlerausgang weitergeleitet.

Dieser Abschnitt erläutert die folgenden Themen:

- *Beispiel: Konvertierung MIME > XML-MIME, S. 190*
- *Dialogbeschreibungen, S. 191*

Verwendung

Ein MIME Adapter konvertiert Nachrichten von MIME nach XML-MIME bzw. von XML-MIME nach MIME:

- **MIME > XML**
  - Wenn das Attribut „charset" im Content-Type einer Nachricht fehlt, dann wird angenommen, dass der Inhalt dieses MIME-Parts mit US-ASCII kodiert ist.
  - Bei Multipart-Nachrichten wird der erste gefundene „charset"-Wert Standard-Zeichensatz für alle weiteren MIME-Parts verwendet, wenn diese keinen eigenen „charset"-Parameter in ihren Content-Types haben.
- **XML > MIME**

  Für die Konvertierung gelten die folgenden Regeln, wenn der Content in der XML-Datei als Text vorliegt statt mit base64 kodiert zu sein:
  - Wenn im XML der Content-Type nicht explizit gesetzt ist, wird `text/plain` angenommen.
  - Wenn der Content-Type im XML gesetzt ist, dann wird die Nachricht mit `charset="UTF-8"` versendet, da es sich nur um Textdaten handeln kann, wenn der Inhalt im XML nicht base64-kodiert vorliegt.

Vorlagen

Zur Unterstützung beim Konvertieren finden Sie folgende Dateien im Repository unter `Global > System > Mapping Templates > MIME Adapter`:

- **`AttachmentExample.mime`**

  Beispielnachricht mit Bild als Anhang
- **`AttachmentExample.xml`**

  Beispielnachricht mit Bildanhang als MIME-XML
- **`IBISMime1.0.xsd`**

  Schema-Datei, die das in der inubit Suite 6 verwendete MIME-XML Format beschreibt und zur Konvertierung MIME>XML-MIME verwendet wird.
- **`Textexample.mime`**

  MIME-Beispielnachricht
- **`Textexample.xml`**

  MIME-Beispielnachricht als MIME-XML

Nachrichtentypen

- **MIME > XML-MIME**: Der Eingangsnachrichtentyp ist unwichtig, da der Nachrichteninhalt MIME-kodiert ist; der Ausgangsnachrichtentyp muss XML sein.
- **XML-MIME > MIME**: Der Typ der Eingangsnachricht muss XML sein, der Ausgangsnachrichtentyp ist unwichtig, da der Nachrichteninhalt MIME-kodiert ist.

9.1 Beispiel: Konvertierung MIME > XML-MIME

Eingangsnachricht MIME-Format

```
Date: Thu, 27 Mar 2003 14:25:15 +0100 (GMT+01:00)
Mime-Version: 1.0
Content-Type: multipart/related;
  boundary="----=_Part_1404_
1807448131.1048771342453"
Content-ID: 1e9aa068-78ac-1000-9e85-c0a80c7a0001

------=_Part_1404_1807448131.1048771342453
Content-Type: plain/text
Content-Transfer-Encoding: 8bit
Content-ID: 1e9aa068-78ac-1000-9e84-c0a80c7a0001
Content-Description: Document
Content-Disposition: inline

This is Part 1
------=_Part_1404_1807448131.1048771342453
Content-Type: plain/text
Content-Transfer-Encoding: 7bit
Content-ID: 1e802c92-78ac-1000-953c-40272c850001
Content-Description: Catalog
Content-Disposition: inline

This is Part 2
------=_Part_1404_1807448131.1048771342453--
```

Ausgangsnachricht XML-MIME

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<IBISMime version="1.0">
  <Header name="Date" value="Thu, 27 Mar 2003
14:25:15 +0100 (GMT+01:00)"/>
```

```
  <Header name="Mime-Version" value="1.0"/>
  <Header name="Content-Type" value="multipart/
related">
    <Parameter name="boundary" value="----=_Part_
1404_1807448131.1048771342453"/>
  </Header>
  <Header name="Content-ID" value="1e9aa068-78ac-
1000-9e85-c0a80c7a0001"/>
  <Header name="Message-ID"
value="&lt;1330082.1053954922869.JavaMail.tf@obelix
.inubit.com&gt;"/>
  <Part>
    <Header name="Content-Type" value="plain/text"/>
    <Header name="Content-Transfer-Encoding"
value="7bit"/>
    <Header name="Content-ID" value="1e9aa068-78ac-
1000-9e84-c0a80c7a0001"/>
    <Header name="Content-Description"
value="Document"/>
    <Header name="Content-Disposition"
value="inline"/>
    <Content>This is Part 1</Content>
  </Part>
  <Part>
    <Header name="Content-Type" value="plain/text"/>
    <Header name="Content-Transfer-Encoding"
value="7bit"/>
    <Header name="Content-ID" value="1e802c92-78ac-
1000-953c-40272c850001"/>
    <Header name="Content-Description"
value="Catalog"/>
    <Header name="Content-Disposition"
value="inline"/>
    <Content>This is Part 2</Content>
  </Part>
</IBISMime>
```

9.2 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „MIME-XML Adaptereigenschaften“, S. 192*
- *Dialog „XML-MIME Adaptereigenschaften“, S. 192*

9.2.1 Dialog „MIME-XML Adaptereigenschaften“

XML Ausgangskonfiguration

- **Header-Namen klein schreiben**
  Erzeugt kleingeschriebene Header-Namen, auf die komfortabel per XPath zugegriffen werden kann.
- **Inhalte immer Base64 kodieren**
  Kodiert den MIME-Inhalt mit dem base64-Verfahren.
  Wenn die Option nicht markiert ist, dann werden Text-Inhalte (MimeType `text/*`) direkt als Textknoten unterhalb von Content geschrieben.
- **Content-Transfer-Encoding möglichst beibehalten**
  Ist diese Option aktiviert, wird versucht, die Kodierung der Eingangsnachricht für den Transport beizubehalten, d. h., Text bleibt Text und mit dem Base64-Verfahren kodierter Inhalt wird unverändert durchgereicht. Eingangsnachrichten mit anderen Kodierungen werden mit Base64 kodiert.

9.2.2 Dialog „XML-MIME Adaptereigenschaften“

Dieser Format Adapter hat keine konfigurierbaren Eigenschaften und kann direkt eingesetzt werden.

Dieser Abschnitt erläutert die folgenden Themen:

- *Eingangsnachrichten für XML-XLS Adapter erstellen, S. 193*
- *XSL > XML: Zeitzone zum Konvertieren von Datumswerten umstellen, S. 194*
- *Dialogbeschreibungen, S. 194*

Verwendung

Der XLS Format Adapter erzeugt aus Microsoft-Excel-Dateien XML-Formate und umgekehrt.

- **XLS > XML**

  Der XLS Format Adapter kann Excel-Dateien im Format XLS der Excel-Versionen 95, 97, 2000, 2003 und 2007 einlesen.

  Sie können einzelne oder alle Arbeitsblätter einer Excel-Datei konvertieren. Auch Excel-typische Darstellungsformate wie Schriftart und Schriftgröße, Farbe, Ausrichtungen werden in der XML-Struktur abgebildet. Für die Zellenformatierung werden Fonts, Datum, Zahlen und arithmetische Formeln unterstützt. String-Funktionen werden nicht unterstützt. Aus allen Formelarten kann der letzte berechnete Wert übernommen werden.

Grafiken können weder eingelesen noch erzeugt werden!

  Das erzeugte XML-Format hat eine dreidimensionale Struktur, die drei Dimensionen entsprechen dem Arbeitsblatt, der Zeile und der Spalte in der Excel-Eingangsnachricht.

- **XML > XLS**

  Beim Konvertieren von XML-Nachrichten in das XLS-Format erwartet der XLS Format Adapter Eingangsnachrichten im XMLSS-Format und gibt Excel-Dateien im Format von Excel 97 aus.

10.1 Eingangsnachrichten für XML-XLS Adapter erstellen

Der XML-XLS Adapter erwartet Eingangsnachrichten im XMLSS-Format. Um Ihre Eingangsnachrichten in dieses Format zu konvertieren, verwenden Sie ein XSLT Converter Modul und das mitgelieferte Mapping Template.

So gehen Sie vor

1. Erstellen Sie einen XSLT Converter.

2. Öffnen Sie im Bereich „XML Zieldatei“ das -Menü und wählen Sie „Öffnen von > Repository“.
   Der Repository Explorer wird angezeigt.
3. Öffnen Sie das Verzeichnis `Global > System > Mapping Templates > XLS Adapter` und markieren Sie die Datei `template.xml`.
4. Klicken Sie auf „OK“. Der Explorer schließt sich und das Template wird angezeigt.
5. Bilden Sie Ihre Eingangsnachricht auf die Template-Struktur ab, um eine Nachricht im erforderlichen Format zu erzeugen.
   → Siehe *XSLT Converter (Data Converter) (Workbench/Process Engine: Modul-Guide, Kap. 5, S. 133)*.

10.2 XSL > XML: Zeitzone zum Konvertieren von Datumswerten umstellen

Beim Konvertieren von Microsoft-Excel-Dateien nach XML können Sie die Zeitzone, die zum Konvertieren von Datumswerten genutzt werden soll, explizit angeben.

Um für Zeitreihen, die sich über den Zeitpunkt der Winter-/Sommerzeitumstellung erstrecken, sicherzustellen, dass die Sommerzeit ignoriert wird und es daher auch einen Wert für 02:00 Uhr gibt, müssen Sie die Zeitzone relativ zur Greenwich Mean Time angeben, z. B. „GMT+1“.

So gehen Sie vor

1. Fügen Sie Ihrem Modul die Moduleigenschaft `timeZone` hinzu.
   → Siehe *Moduleigenschaften hinzufügen (Workbench: Benutzer-Guide, Kap. 3.7, S. 123)*.
2. Geben Sie als Wert das Kürzel der Zeitzone ein, die für Ihre Daten in der Excel-Datei gilt.

10.3 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „XLS-XML Adapter Eigenschaften“, S. 195*
- *Dialog „XML-XLS Adapter Eigenschaften“, S. 196*

10.3.1 Dialog „XLS-XML Adapter Eigenschaften“

Dieser Dialog bietet folgende Optionen:

Auswahl des XLS-Prozessors

- **Prozessor**
  Wählen Sie den für Ihre Eingangsdaten geeigneten XLS- bzw. XLSX-Prozessor.

Auswahl Arbeitsblätter

- **Arbeitsblatt**
  - **Alle**
    Um alle Arbeitsblätter einer Excel-Datei einzulesen.
  - **Einzelnes**
    Um ein ausgewähltes Arbeitsblatt einzulesen.
    Über den Button „Auswählen“ können Sie dazu eine Excel-Datei laden. Die darin enthaltenen Arbeitsblätter werden in der Auswahlbox angezeigt.

Zeilen und Spalten

- **Leere Zeilen mit aufnehmen**
  Sollen Leere Zeilen, die vor anderen Zeilen mit Daten stehen, im erzeugten XML-Format als leere XML-Tags übernommen werden, so wählen Sie diese Checkbox. Anderenfalls werden solche leeren Zeilen ignoriert. Leere Zeilen am Ende eines Arbeitsblattes werden immer ignoriert.
- **Leere Zellen mit aufnehmen**
  Sollen leere Zellen, die vor anderen Zellen mit Daten stehen, im erzeugten XML-Format als leere XML-Tags übernommen werden, so wählen Sie diese Checkbox. Anderenfalls werden solche leeren Zellen ignoriert. Leere Zellen am Ende einer Zeile werden immer ignoriert.
- **Zeilen am Anfang überspringen**
  Anzahl der Zeilen, die ignoriert werden sollen.
- **Spalten am Anfang überspringen**
  Anzahl der Spalten, die in jeder Zeile ignoriert werden sollen.

XML Spreadsheet (XMLSS) Konfiguration

- **Zahlendarstellung**
  - **Exponentialdarstellung**
    Repräsentiert die Zahlen in Exponentialdarstellung
  - **Dezimaldarstellung**
    Wenn Sie die Dezimaldarstellung wählen, müssen Sie als nächstes das Trennzeichen angeben.
- **Dezimaltrennzeichen**
  Wählen Sie zwischen Punkt und Komma
- **Mindestanzahl Nachkommastellen**
  Anzahl der Nachkommastellen.
- **Mit XML Namensräume**
  - Wenn markiert: XML wird mit Namensräumen erstellt.
  - Wenn nicht markiert: XML wird ohne Namensräumen erstellt.
- **Layout-Informationen ins XML übertragen**:
  Wenn markiert, dann werden die Angaben zu verwendeten Farben, Linien und Schriftstärken etc. ins XML übernommen, ansonsten werden diese verworfen. Die Übernahme von Layout-Informationen erhöht die Größe der XML-Datei beträchtlich.
- **Datumsformat**
  Wählen Sie ein Datumsformat aus.
- **Zeichensatz**
  Die Standardzeichensatzkodierung von Excel ist UTF-8. Darüber hinaus sind die Zeichensätze US-ASCII, ISO-8859-1, UTF-16BE, UTF-16LE und UTF-16 wählbar. Die sechs Zeichensätze sind vorausgewählt, da die Firma SUN, welche die Java Entwicklungs- und Laufzeitumgebung erstellt, garantiert, dass diese Zeichensätze immer unterstützt werden.
  Sie können auch jeden anderen beliebigen Zeichensatz eintragen.

10.3.2 Dialog „XML-XLS Adapter Eigenschaften“

Dieser Dialog bietet folgenden Optionen:

Auswahl des XLS-Prozessors

Wählen Sie den für Ihre Ausgangsdaten geeigneten XLS- bzw. XLSX-Prozessor.

Das folgende Schema enthält alle Elemente und Attribute, die bezogen auf POI 37 unterstützt werden: `Global/System/Mapping Templates/XLS Adapter/POI37.xsd`

XML Spreadsheet (XMLSS) Konfiguration

- **Datumsformat**
  Wählen Sie ein Datumsformat aus.

Aktion bei fehlendem Dokument

Legen Sie fest, was passieren soll, falls die XML-Datei leer ist:

- **XSL-Dokument mit leerer Tabelle erzeugen**
- **Fehlermeldung erzeugen**

Dieser Abschnitt erläutert die folgenden Themen:

- *Eingangsnachrichten für die Konvertierung von XML > XLSX erstellen, S. 199*
- *Dialog „XLSX Adapter Eigenschaften“, S. 200*

Verwendung

Das XLSX-Format (ab Office 2007) ist ein Zip-Archiv mit verschiedenen XML-basierten Konfigurations- und Datendateien.

Der XLSX Format Adapter überführt dieses Archiv in eine einzelne XML-Struktur basierend auf dem IBISDirectory-XML-Format bzw. erzeugt aus einer Datei im IBISDirectory-XML-Format ein XLSX-Zip-Archiv.

Bei der Konvertierung von XLSX > XML werden keine Layoutangaben übernommen.

11.1 Eingangsnachrichten für die Konvertierung von XML > XLSX erstellen

Bei der Konvertierung von XML > XLSX erwartet der XLSX Adapter Eingangsnachrichten im IBISDirectory-XML-Format.

Um Ihre Eingangsnachrichten in dieses Format zu konvertieren, verwenden Sie einen XSLT Converter und das mitgelieferte Mapping Template.

So gehen Sie vor

1. Erstellen Sie einen XSLT Converter.
2. Öffnen Sie im Bereich „XML Zieldatei“ das -Menü und wählen Sie „Öffnen von > Repository“. Der Repository Explorer wird angezeigt.
3. Öffnen Sie das Verzeichnis `Global > System > Mapping Templates > XLSX Adapter` und markieren Sie die Datei `ExampleFolder.xml`
4. Klicken Sie auf „OK“. Der Explorer schließt sich und das Template wird angezeigt.
5. Bilden Sie nun Ihre Eingangsnachricht auf die Template-Struktur ab.

   → Siehe *XSLT Converter (Data Converter) (Workbench/Process Engine: Modul-Guide, Kap. 5, S. 133).*

11.2 Dialog „XLSX Adapter Eigenschaften“

Grundkonfiguration

- **Konvertierungsrichtung**
  Wählen Sie eine der beiden Konvertierungsrichtungen aus:
  - XLSX nach IBISDirectory-XML
  - IBISDirectory-XML nach XLSX

Dieser Abschnitt erläutert die folgenden Themen:

- *Beispiel-Workflow, S. 201*
- *Dialog „Barcode Generator Eigenschaften“, S. 203*

Verwendung

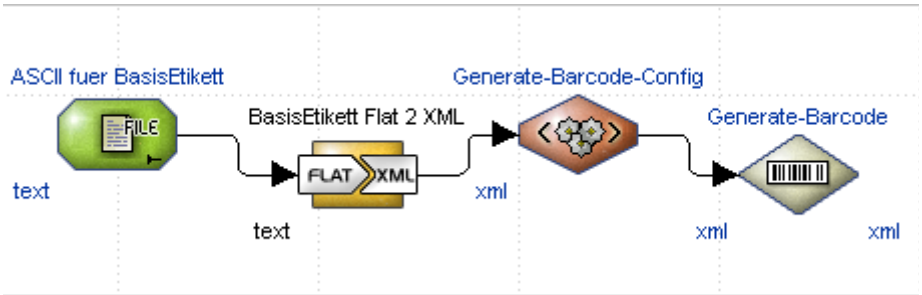
Ein Barcode Generator erzeugt aus einem alphanumerischen Wert einen Barcode.

Jedes Barcode Generator Modul kann genau eine Barcode-Art erzeugen; um verschiedene Barcode-Arten zu erzeugen, müssen Sie mehrere Barcode Generator Module verwenden.

Der alphanumerische Wert, aus dem der Barcode erzeugt wird, muss in einer XML-Eingangsnachricht an das Modul übergeben werden. Die Art des zu erzeugenden Barcodes legen Sie beim Konfigurieren des Barcode Generator Moduls fest.

Der Barcode wird als SVG-File in einer XML-Nachricht ausgegeben, aus der z. B. mit dem FO Converter Modul eine PDF-Datei erzeugt werden kann.

12.1 Beispiel-Workflow



Der abgebildete Workflow erzeugt aus einer Artikelnummer in einer Text-Datei einen Barcode.

ASCII > XML

Der File Connector sendet die Text-Datei an einen Flat-XML-Adapter. Der Flat-XML-Adapter erzeugt aus der flachen Textstruktur eine einfache XML-Struktur, aus

```
157560 | 40684 | 2700928058BJ | 64130037058
```

wird

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Flatfile>
```

```
  <NS>
    <ProduktionsAuftrag>157560</ProduktionsAuftrag>
    <Trennz1>|</Trennz1>
    <Projekt>40684</Projekt>
    <Trenn2>|</Trenn2>
    <Artikel>2700928058BJ</Artikel>
    <Trenn3>|</Trenn3>
    <ArtikelNrKunde>64130037058</ArtikelNrKunde>
  </NS>
</Flatfile>
```

XML > XML

Der XSLT Converter erzeugt aus dem XML das folgende, spezielle XML-Format, welches der Barcode Generator erwartet:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Barcodes>
  <Barcode>
    <ID>Kundenartikelnummer</ID>
    <!-- war: ArtikelNrKunde -->
    <Value>64130037058</Value>
  </Barcode>
</Barcodes>
```

Das XML-Barcode-Schema finden Sie im Repository unter `Global > System > Mapping Templates > Barcode Utility > barcode.xsd`.

XML > Barcode XML

Der Barcode Generator erzeugt aus dem Inhalt des Elements Value einen SVG-kodierten Barcode:

../../ibis_root\root\tmp\40_BasisEtikett_Drucken_29112007_165127_WThread-0
Datei ansehen...
Im oberen Bereich wird die Ergebnisdatei angezeigt, im unteren Bereich werden die Workflow Variab
Barcodes
Barcode
ID : Kundenartikelnummer
Value : 64130037058
svg (height='13mm' width='39.425mm' xmlns:='http://www.w3.org/2000/svg')
g (style='fill:black; stroke:none')
rect (height='10.1781mm' width='0.19mm' x='1.9mm' y='2.8219mm')
rect (height='10.1781mm' width='0.19mm' x='2.565mm' y='2.8219mm')
rect (height='10.1781mm' width='0.475mm' x='2.945mm' y='2.8219mm')
rect (height='10.1781mm' width='0.475mm' x='3.61mm' y='2.8219mm')
rect (height='10.1781mm' width='0.19mm' x='4.275mm' y='2.8219mm')
rect (height='10.1781mm' width='0.19mm' x='4.655mm' y='2.8219mm')

Aus diesem XML kann nun unter Einsatz eines weiteren XSLT Converters und eines FO Converters z. B. ein PDF erzeugt werden.

12.2 Dialog „Barcode Generator Eigenschaften“

In diesem Dialog legen Sie die Art des zu erzeugenden Barcodes und dessen Layout fest.

Sobald Sie eine Barcode-Art auswählen, wird im Dialog eine Erläuterung dazu angezeigt und die Eingabe- und Auswahlfelder werden sinnvoll vorbelegt.

Die angezeigten Eingabe- und Auswahlfelder zum Definieren des Layouts sind abhängig von der Art des ausgewählten Barcodes.

Längenangaben sind in mm oder mw (Verhältnis zur Modulbreite) anzugeben.

Dieser Abschnitt erläutert die folgenden Themen:

- *Zip-Archiv mit mehreren Dateien erstellen, S. 205*
- *Dialog „Compressor/Decompressor Eigenschaften“, S. 206*

Verwendung

Mit dem Compressor können Sie eingehende Nachrichten komprimieren bzw. bereits komprimierte Nachrichten entpacken.

13.1 Zip-Archiv mit mehreren Dateien erstellen

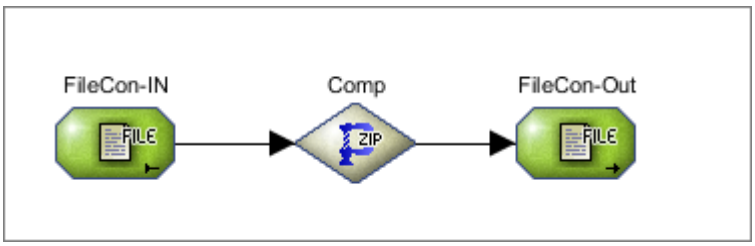
Sie können mit einem Compressor mehrere Dateien komprimieren.

Voraussetzungen

Die Dateien sind in einem Verzeichnis im Dateisystem gespeichert.

So gehen Sie vor

1. Legen Sie folgenden Technical Workflow an:



   Der File Input Connector liest das Verzeichnis ein, der Compressor komprimiert das Verzeichnis und der File Output Connector schreibt das Zip-Archiv ins Dateisystem.
2. Konfigurieren Sie den File Input Connector wie folgt:
   - Der Verzeichnismodus ist ausgewählt.
   - Datenweitergabe=IBISDirectory-Xml
   - Bei „XML-Konfiguration“ ist Kodierung=UTF-8 ausgewählt und die Optionen „inklusive Daten“ und „Daten base64 kodieren“ sind markiert.
3. Wählen Sie beim Compressor folgende Optionen:
   - Verfahren=Zip
   - Eingangsformat=IBISDirectory XML
4. Wählen Sie im File Output Connector als Eingangsformat „Daten“ aus.

13.2 Dialog „Compressor/Decompressor Eigenschaften“

In diesem Dialog haben Sie folgende Optionen:

Einstellungen

- **Aktion**
  Wählen Sie eine der folgenden Aktionen:
  - **Daten komprimieren**
    Die Eingangsnachricht wird mit dem gewählten Verfahren gepackt.
  - **Daten dekomprimieren**
    Die Eingangsnachricht wird entpackt.
    Wenn die Eingangsnachricht nicht als Zip/GZip identifiziert werden oder nicht entpackt werden kann, dann wird ein Fehler geworfen.
- **Verfahren**
  - **ZIP**
    Zum Komprimieren mehrerer Dateien und von Verzeichnisbäumen.
  - **GZIP** (GNUzip)
    Zum Komprimieren einzelner Dateien.

Wenn beim GZIP-Verfahren in der Datei ein Dateiname und/ oder ein Datei-Änderungsdatum gefunden wird, werden diese ausgelesen und als Variablen `ReadFileName` und `ReadFileDate` ausgegeben.

- **Eingangsformat** (bei Aktion=Daten komprimieren)
  - **Daten**
    Zu markieren für alle Dateiformate außer IBISDirectoryXML.
  - **IBISDirectory XML**
    Markieren Sie diese Option, wenn die Eingangsnachricht im IBISDirectory XML-Format vorliegt. Nur dann kann die Nachricht korrekt komprimiert werden.
- **Ausgangsformat** (wenn Aktion=Daten dekomprimieren)
  - **Daten**
    Zu markieren für alle Dateiformate außer IBISDirectoryXML.
  - **IBISDirectory XML**
    Markieren Sie diese Option, wenn die Nachricht im IBISDirectory XML-Format ausgegeben werden soll. Nur dann kann die Nachricht korrekt entpackt werden.

Zip-Optionen

(nur bei Verfahren=zip)

- **Kodierung des Dateinamens**

Die Kodierung des Dateinamens muss mit der Kodierung des Archivs übereinstimmen. Wenn die beiden Kodierungen nicht übereinstimmen, dann werden u. U. Dateinamen nicht gefunden, die Zeichen außerhalb des US-ASCII-Charakterzeichensatzes (Umlaute etc.) enthalten.

- **Dateiname innerhalb des Zip-Archivs**
  - Beim Komprimieren:
    Name der Datei im Zip-Archivs, welches der Compressor erzeugt. Wenn Sie keinen Namen angegeben, wird standardmäßig „IBISData“ verwendet.
  - Beim Dekomprimieren:
    Name und Endung der Datei, die entpackt werden soll.
- **Kommentar im Zip-Archiv**
  Kommentar.

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialogbeschreibungen, S. 209*

Verwendung

Mit dem Cryptographer können Sie Daten verschlüsseln, entschlüsseln und signieren.

14.1 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Cryptographer Plug-in“, S. 209*
- *Dialog „Websale Cryptographic 3DES“, S. 210*
- *Dialog „CMS (Cryptographic Message Syntax)“, S. 210*

14.1.1 Dialog „Cryptographer Plug-in“

In diesem Dialog wählen Sie eines der angebotenen Verfahren aus.

Verfügbare Verfahren

- **Websale Cryptographic 3DES**
  Symmetrischer Verschlüsselungsalgorithmus für den Nachrichtenaustausch.
  Bei diesem Verfahren wird ein und derselbe Schlüssel zum Verschlüsseln und zum Entschlüsseln der Eingangsnachricht genutzt.
- **CMS (Cryptographic Message Syntax)**
  Wenn Sie diese Option wählen, wird die Eingangsnachricht in einem hybriden Verfahren verschlüsselt/entschlüsselt und übermittelt. Das hybride Verfahren kombiniert die asymmetrischen Verschlüsselung/Entschlüsselung mit der symmetrischen:
  - Zur Laufzeit wird automatisch ein Session Key generiert. Mit diesem Session Key wird die Eingangsnachricht symmetrisch verschlüsselt. Welches symmetrische Verfahren dazu genutzt wird, legen Sie im nächsten Schritt fest.

- Der Session Key und die Eingangsnachricht werden mit dem asymmetrischen Verfahren „CMS“ an den Empfänger übermittelt.

Damit nutzt der Cryptographer die Vorteile beider Verfahren: symmetrische Verfahren sind bei großen Datenmengen performanter, asymmetrische Verfahren sind sicherer.

- **Unix Passwort Hashing (crypt)**
  Zum Signieren der Eingangsnachricht mit einem alphanumerischen Wert.
  Damit kann überprüft werden, ob die Eingangsnachricht z. B. bei der Übertragung über ein unsicheres Netz verfälscht wurde, da das Verfahren stets denselben Wert liefert, wenn es auf dieselbe Eingangsnachricht angewendet wird.

14.1.2 Dialog „Websale Cryptographic 3DES“

Wenn Sie im *Dialog „Cryptographer Plug-in“ (Workbench/Process Engine: Modul-Guide, Kap. 14.1.1, S. 209)* „Websale Cryptographic 3DES“ als Ver-/Entschlüsselungsmethode gewählt haben, dann legen Sie in diesem Dialog, ob der Cryptographer die Eingangsnachricht verschlüsseln oder entschlüsseln soll.

Einstellungen

- **Verschlüsseln/Entschlüsseln**
  Wählen Sie aus, ob die Eingangsnachricht ver- oder entschlüsselt werden sollen.
- **Passwort/Passwort wiederholen**
  Die angegebene Zeichenkette wird als Schlüssel für die symmetrische Ver-/Entschlüsselung verwendet.

Wenn Sie Nachrichten verschlüsseln, muss der Empfänger der Nachrichten den Schlüssel kennen, um diese entschlüsseln zu können!

14.1.3 Dialog „CMS (Cryptographic Message Syntax)“

Wenn Sie im *Dialog „Cryptographer Plug-in“ (Workbench/Process Engine: Modul-Guide, Kap. 14.1.1, S. 209)* „CMS“ als Ver-/Entschlüsselungsmethode gewählt haben, dann geben Sie in diesem Dialog an, ob der Cryptographer die Eingangsnachricht und den

Session Key verschlüsseln oder entschlüsseln soll und welche symmetrische Methode dazu genutzt werden soll. Zusätzlich hinterlegen Sie Ihre öffentlichen bzw. privaten Schlüssel.

Verschlüsselungseinstellungen

- **Verschlüsseln**
  Legt fest, dass der Cryptographer die Eingangsnachricht und den Session Key verschlüsseln soll.
- **Zertifikat oder Truststore auswählen…**
  Zum Hinterlegen des Schlüssels, mit dem die Eingangsnachricht und der Session Key verschlüsselt werden.
  Nach Auswahl wird die Gültigkeit des Zertifikats bzw. Truststores angezeigt.
- **Verschlüsselungsalgorithmus**
  Zur Auswahl des symmetrischen Verfahrens, mit dem die Eingangsnachricht und der Session Key verschlüsselt werden.

Entschlüsselungseinstellungen

- **Entschlüsseln**
  Legt fest, dass der Cryptographer die Eingangsnachricht und den Session Key entschlüsseln soll.
- **Privaten Schlüssel auswählen…**
  Zum Hinterlegen des Schlüssels, mit dem die Eingangsnachricht und den Session Key entschlüsselt werden.

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Data Stream Modifier Eigenschaften“, S. 213*

Verwendung

Mit dem Data Stream Modifier können Sie Vorkommen bestimmter Zeichen aus den Eingangsnachrichten durch andere Zeichen ersetzen und so z. B. Zeilenumbrüche entfernen.

Ein Data Stream Modifier kann alle Nachrichtentypen verarbeiten. Sie können ASCII-, Hexadezimal- oder Oktal-Werten suchen sowie den Bereich einschränken, in dem gesucht und ersetzt werden soll.

15.1 Dialog „Data Stream Modifier Eigenschaften“

Verarbeitungsbereich einschränken

Wenn die Option markiert ist, kann ein Bereich in der Datei definiert werden, in dem die Ersetzungen durchgeführt werden sollen:

- **Von Position**: Beginn des Bereichs.
- **Anzahl Bytes:** wird automatisch berechnet
- **Bis Position**: Ende des Bereichs. Wird berechnet aus der Anfangsposition und der Anzahl Bytes.

Einstellungen

Es können mehrere Ersetzungspaare definiert sein. Die Ersetzungen werden in der Reihenfolge durchgeführt, wie sie in der Tabelle angezeigt werden.

Die Buttons über der Tabelle dienen zum Anlegen, Löschen und Verschieben der Ersetzungspaare.

Zeicheneingabe

Zum Eingeben des zu ersetzenden Wertes bzw. des Ersatzwertes. Doppelklicken Sie in der Spalte „Suchen nach“ bzw. „Ersetzen mit“ in eine Tabellenzelle und in der Zelle auf . Ein Dialog wird angezeigt.

Geben Sie den zu ersetzenden Wert in eine der Zeilen ein. Die passenden anderen Zeichenrepräsentationen werden automatisch gebildet.

Um festzulegen, wie das Zeichen in der Tabelle angezeigt werden soll, markieren Sie eine Zeichenrepräsentation, z. B. Integer.

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Encoder Eigenschaften“ , S. 215*

Verwendung

Mit dem Encoder-Utility können Sie Eingangsnachrichten dekodieren, kodieren und dabei z. B. mit einer Prüfsumme versehen oder den Zeichensatz ändern:

- Das Kodieren von Daten ist z. B. nötig, wenn die Daten für E-Mail-Übertragungen oder XML-Darstellungen in einer ASCII-basierten Darstellung vorliegen müssen.
- Beim Dekodieren werden die Daten wieder in ihr ursprüngliches Format zurückgeführt.

16.1 Dialog „Encoder Eigenschaften“

In diesem Dialog legen Sie die Funktionsweise des Encoder-Utilitys fest.

Konfiguration Eingangsdaten

- **Eingangsdaten aus XML**
  Wenn die Eingangsdaten XML-formatiert sind, dann können Sie auch einzelne Elemente kodieren oder dekodieren lassen: Geben Sie den XPath zu dem XML-Element ein, dessen Inhalt kodiert/dekodiert werden soll. Wenn Sie eine Beispiel-Eingangsdatei haben, klicken Sie auf den Button, um die Datei zu laden und zu dem XML-Element zu navigieren. Der XPath wird dabei automatisch erstellt.
- **Zeichensatzkodierung**
  Nur aktiv, wenn als Algorithmus „Zeichensatzkodierung“ oder „URL“ gewählt ist. Wählen Sie eine Zeichensatzkodierung aus der Liste aus.

Operation

- **Aktion wählen**
  Wählen Sie aus, ob die Eingangsdaten kodiert oder dekodiert werden sollen.
- **Algorithmus wählen**
  Die folgenden Verfahren stehen zur Auswahl:
  - **Zeichensatzkodierung**: Wenn diese Option ausgewählt, dann können Sie eine Zeichensatzkodierung für die Ausgangsdaten auswählen.

- **base64**: Die base64-Kodierung wird benutzt, um Binärdaten über Technologien zu übertragen, die 8-Bit nicht sauber übertragen können, wie z. B. E-Mail.

  Die binären Zeichenketten werden in Text-Zeichenketten umgewandelt, die problemlos per E-Mail versendet oder als Teil eines HTTP POST Requests verwendet werden können. base64 verwendet eine Untermenge des ASCII Zeichensatzes, die kodierten Zeilen müssen exakt 64 druckbare Zeichen lang sein. Durch die Kodierung steigt der Platzbedarf der Daten um 25%.
- **MIME-base64**: Dieses Verfahren basiert auf der base64-Kodierung, legt aber keine feste Länge für base64-kodierte Zeilen fest. Lediglich die Maximallänge ist mit 76 Zeichen fest definiert.
- **hex**: Dieser Algorithmus wandelt Binärdaten in eine Folge von Hexadezimal-Zahlen um.
- **URL**: URL-Encoding ist notwendig, wenn Eingangsnachrichten als HTTP Request oder HTTP Response gesendet werden soll. Da bei der HTTP Kommunikation nur eine Untermenge des US-ASCII Zeichensatz verwendet werden darf, müssen alle Zeichen, die nicht in dieser Untermenge enthalten sind, durch sog. Escape-Sequenzen ersetzt werden. So wird z. B. das Leerzeichen durch die Sequenz%20 ersetzt.

Mit den folgenden Algorithmen können Sie **Prüfsummen** erstellen lassen, mit denen die Integrität von Nachrichten geprüft werden kann. Um festzustellen, ob eine Nachricht verändert oder beschädigt wurde, wird die aktuelle Prüfsumme der Nachricht mit einer bekannten, früher erstellten Summe verglichen. Die Prüfsumme wird am Ende der Nachricht übertragen. Zur Auswahl stehen die folgenden Algorithmen:

- **MD5**: Eine weit verbreitete kryptographische Funktion, die aus beliebigen Daten eine 128-Bit-Prüfsumme erzeugt. MD5-Summen werden unter anderem von PGP verwendet.
- **SHA1**: Kryptographische Funktion, die aus beliebigen Daten eine 160-Bit-Prüfsumme erzeugt.
- **SHA-224**, **SHA-256**, **SHA-384** und **SHA-512**: Kryptographische Funktionen, die zur SHA-2-Familie der Hash-Algorithmen gehören und durch bessere Algorithmen das Kollisionsrisiko reduzieren.

Konfiguration Ausgangsdaten

Zeichensatzkodierung

Nur aktiv, wenn als Aktion „Kodieren“ und als Algorithmus „Zeichensatzkonvertierung“ gewählt ist.

Wählen Sie eine Zeichensatzkodierung aus der Liste aus, um anzugeben, mit welchem Zeichensatz die Ausgangsdaten ausgegeben werden sollen.

Dieser Abschnitt erläutert die folgenden Themen:

- *Funktionsprinzip, S. 218*
- *Eingangsnachricht zum Aufrufen der Operationen erstellen, S. 219*
- *Verschlüsselte Login-Daten verwenden, S. 220*
- *Operationen remote/als anderer Benutzer ausführen, S. 221*
- *Logeinträge aus Monitoring-Komponenten lesen, S. 221*
- *Logeinträge filtern, S. 224*
- *Logeinträge sortieren, S. 225*
- *In Logeinträgen nach Zeichenketten suchen, S. 226*
- *Größe von Eingangsnachrichten ermitteln, S. 226*

Verwendung

Mit dem inubit IS Configuration Utility können Sie während einer Workflow-Ausführung auf Funktionalitäten der inubit Workbench zugreifen. Sie können u. a. folgende Operationen aufrufen:

- Speicher der JVM freigeben bzw. Größe des freien Speicherplatzes auslesen
- Gültigkeit von Plug-in-Lizenzen überprüfen
- Einträge im Connection Manager erzeugen und löschen
- Prozesse aus dem Queue Manager löschen
- Benutzer erzeugen und bearbeiten (z. B. Passwort ändern)
- Technical Workflows kopieren, löschen, exportieren, importieren, aktivieren, bearbeiten (aktivieren/deaktivieren, Metadaten ändern)
- Eigenschaften eines Technical Workflows auslesen

  (u. a. Name, Kommentar, Status aktiv/inaktiv, verwendete Modultypen, Besitzer, Versionsnummer, Datum und Urheber der letzten Änderungen, Workflowgruppe)
- Eigenschaften von Modulen lesen und ändern
- Organigramme lesen
- Konfiguration des Portalservers lesen
- alle sichtbaren Benutzer des Portalservers auslesen
- Taskliste von Benutzern lesen
- Liste aller Benutzer lesen, die für einen ausgewählten Benutzer sichtbar sind
- Liste aller Workflows lesen, die für den aktuellen Benutzer bzw. die aktuelle Benutzergruppe sichtbar sind
- Zum Debuggen den Status und das Level in Server-Traces ändern
- Logs auslesen

  → Siehe *Logeinträge aus Monitoring-Komponenten lesen (Workbench/Process Engine: Modul-Guide, Kap. 17.5, S. 221).*

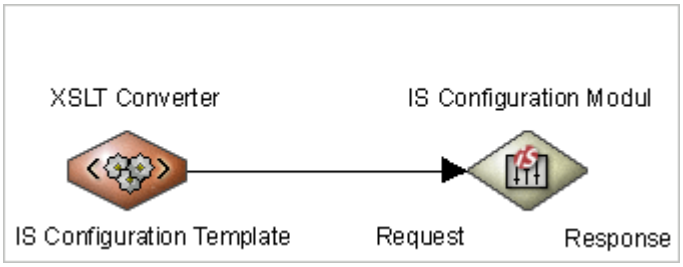
Eine vollständige Übersicht über alle Operationen des inubit IS Configuration Utilitys finden Sie im Configuration Mapping Template im Repository. Siehe auch *Funktionsprinzip (Workbench/Process Engine: Modul-Guide, Kap. 17.1, S. 218)*.

17.1 Funktionsprinzip

Das IS Configuration Utility erwartet als Eingangsnachricht einen XML-Request. Mit diesem XML-Request greift das Utility lesend oder schreibend auf die Konfiguration der inubit Process Engine zu und gibt das Ergebnis als Ausgangsnachricht aus.

Request erstellen

Den XML-Request erstellen Sie mit Hilfe eines XSLT Converters, in den Sie das Configuration Mapping Template laden. Das Mapping Template enthält alle Operationen des inubit IS Configuration Utilitys. Aus dem Template wählen Sie die gewünschten Operationen aus und passen diese an. Sie können in jedem XML-Request beliebig viele verschiedene Operationen ausführen lassen.



→ Siehe *Eingangsnachricht zum Aufrufen der Operationen erstellen (Workbench/Process Engine: Modul-Guide, Kap. 17.2, S. 219).*

Verhalten im Fehlerfall

Wenn eine der im Request definierten Operationen nicht ausgeführt werden kann, dann wird die Ausführung mit der nachfolgenden Operation fortgesetzt.

Inhalt der Response

Das IS Configuration Utility gibt als Ergebnis eine XML-Response aus, die neben dem XML-Request auch Statusinformationen über die ausgeführte Operation enthält.

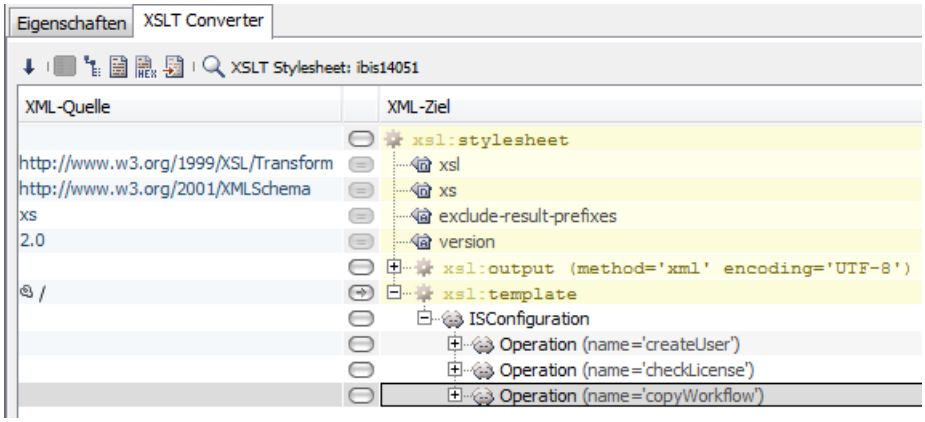
Die XML-Response können Sie z. B. in einem weiteren XSLT Converter verarbeiten.

17.2 Eingangsnachricht zum Aufrufen der Operationen erstellen

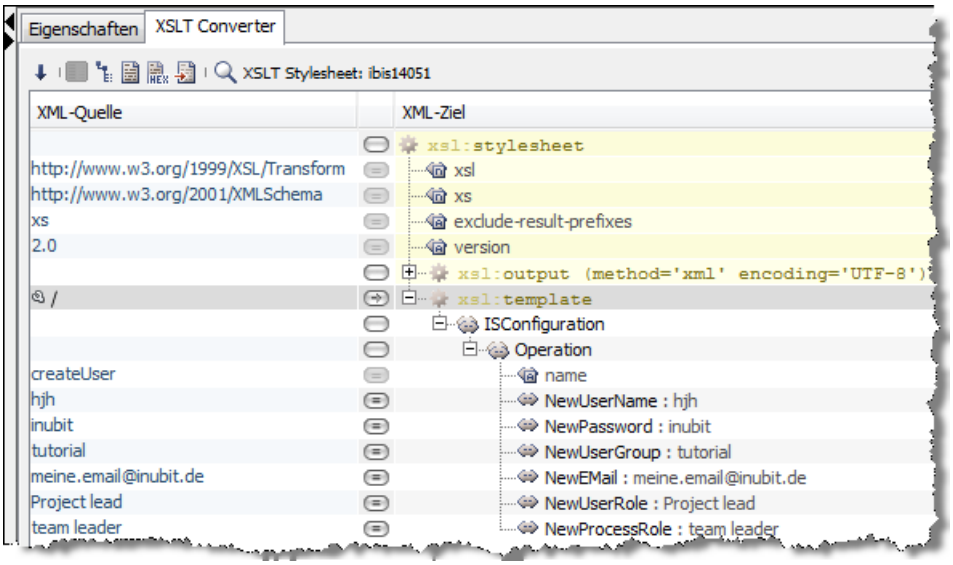
So gehen Sie vor

1. Öffnen Sie im XSLT Converter im Bereich „XML-Zieldatei“ das Pfeilmenü.
2. Wählen Sie „Öffnen von…> Repository“. Der Repository Explorer wird angezeigt.
3. Öffnen Sie den Ordner „Global System > Mapping Templates > IS Configuration“.
4. Markieren Sie das Template `configuration.xml`.
5. Klicken Sie auf „OK“. Der Explorer schließt sich. Das Template wird angezeigt. Es enthält für alle Operationen kommentierte XML-Strukturen in folgender Form:

6. Um eine Operation in einem XSLT-Skript zu verwenden, ziehen Sie das `ISConfiguration`-Element, das zu der entsprechenden Operation gehört, auf das `xsl:template`-Element im XSLT Converter-Skriptbereich.
7. Um weitere Operationen hinzuzufügen, fügen Sie die entsprechenden `operation`-Elemente unterhalb des `ISConfiguration`-Elements ein:



8. Bei einigen Operation müssen Sie Werte für Elemente angeben, z. B. beim Anlegen eines Benutzers den Namen, das Passwort und die Benutzergruppe:



17.3 Verschlüsselte Login-Daten verwenden

Sie können verschlüsselte Login-Daten verwenden, indem Sie dem Modul „inubit IS Configuration“ zwei Moduleigenschaften hinzufügen und diese über das Variablen-Mapping (Variablentyp: `is:password`) setzen.

So gehen Sie vor

1. Fügen Sie den Moduleigenschaften die Eigenschaften `Password` und `Login` mit dem Datentyp `Text (einzeilig)` hinzu.
   → Siehe *Moduleigenschaften hinzufügen (Workbench: Benutzer-Guide, Kap. 3.7, S. 123)*.
   In der Eigenschaftentabelle werden die Namen `Password` und `Login` angezeigt.
2. Speichern Sie Ihre Angaben.

17.4 Operationen remote/als anderer Benutzer ausführen

Standardmäßig werden alle Operationen lokal mit der Berechtigung des Workflow-Eigentümers ausgeführt. Sie können diese aber auch auf einer anderen, entfernten inubit Process Engine und unter der Berechtigung eines anderen Benutzers ausführen lassen.

So gehen Sie vor

Fügen Sie die Elemente `User`, `Password` und `ServerURL` in Ihr XSLT Stylesheet unterhalb des `ISConfiguration`-Elements ein.

Geben Sie die Werte des Benutzers ein, unter dessen Berechtigung die Operationen ausgeführt werden sollen, sowie die URL der inubit Process Engine, auf dem das Modul ausgeführt wird.

Sie können die Elemente per Drag'n'Drop aus dem Configuration-Template auf Ihr XSTL-Stylesheet ziehen.

17.5 Logeinträge aus Monitoring-Komponenten lesen

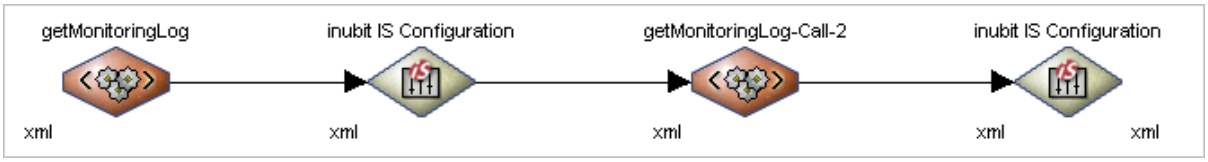
Mit Hilfe eines IS Configuration Moduls können Sie die Log-Dateien auslesen, die in den folgenden Monitoring-Komponenten angezeigt werden:

- System Log
- Connection Manager
- Queue Manager
- Scheduler Manager
- Audit Log

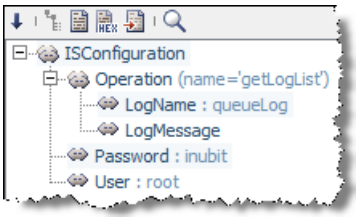
- Process Log

So gehen Sie vor

1. Für eine Anfrage an eine Monitoring-Komponente benötigen Sie immer einen Technical Workflow mit mehreren Modulen, z. B.

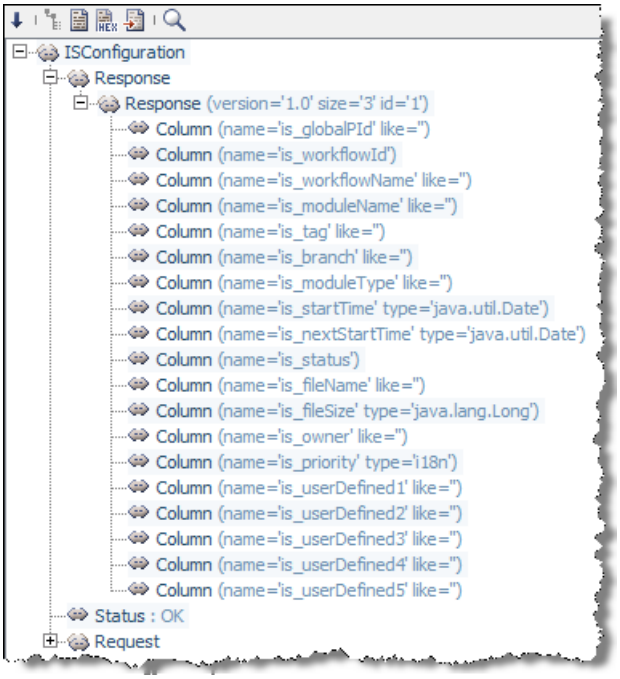


2. Mit Hilfe des ersten XSLT Converters erzeugen Sie einen XML-Request, um die Tabellenstruktur einer bestimmten Monitoring-Komponente abzufragen, z. B.:



Als Wert des Elements `LogName` müssen Sie die internen Namen der Komponenten verwenden; diese finden Sie im Configuration-Template!

Das IS Configuration Modul gibt eine XML-Response mit Infos über die Tabellenstruktur aus, z. B.:

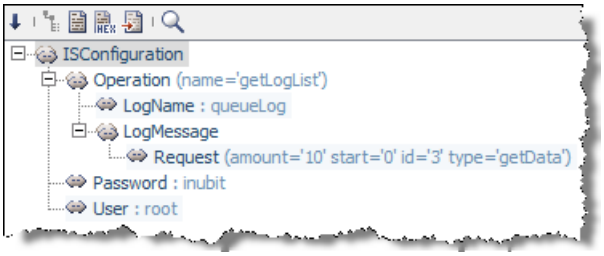


Dabei hat die XML-Response immer folgende Struktur:

- `Response`-Element mit Anzahl der Tabellenspalten (`size`-Attribut), und einem generierten, eindeutigen Bezeichner für diese Response (`ID`-Attribut).

- `Column`-Elemente mit Namen der Spalten (`name`-Attribut), Typ (`type`-Attribut) und Angabe, ob eine Textsuche in der Spalte möglich ist (`like`-Attribut)
- `Status`-Element mit Angabe über den Erfolg der Response
- `Request`-Element mit dem ursprünglichen Request

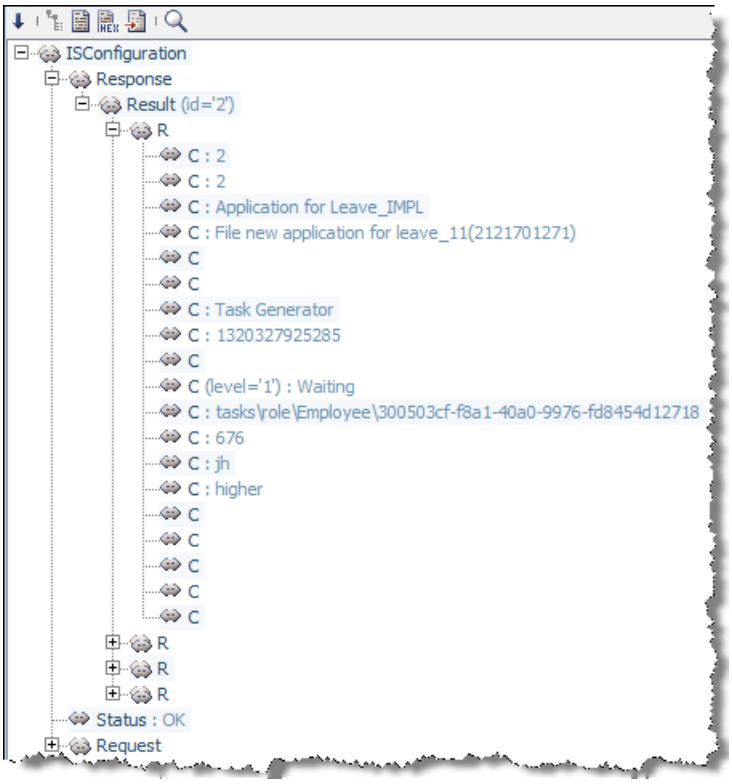
3. Mit dem nächsten XSLT Converter erzeugen Sie auf Basis der Informationen aus der Response den eigentlichen Request, mit dem hier z. B. die ersten 10 Datensätze aus dem Queue Manager gelesen werden sollen:



Dabei haben die Attribute folgende Bedeutungen:

- `type`: Liest alle Werte der Tabelle (`getData`) oder alle Werte einer bestimmten Tabellenspalte, die als Wert des Attributs `restrict` angegeben ist (`getAvailable`)
- `restrict`: Name der Spalte, wenn nur die Werte aus einer Spalte gelesen werden sollen
- `id`: Eindeutiger Bezeichner der Response, auf die der Request Bezug nimmt
- `start`: Erster Datensatz in der Tabelle, der gelesen werden soll; gültig sind numerische Werte oder XPath-Ausdrücke
- `amount`: Anzahl der Datensätze, gelesen werden sollen, beginnend mit dem Datensatz, der durch das Attribut `start` festgelegt wurde; gültige sind numerische Werte oder XPath-Ausdrücke

4. Daraufhin erzeugt das letzte IS Configuration Modul die Response mit den gesuchten Daten:



Das `Result`-Element enthält die gesuchten Datensätze, dabei steht `R` (row) für einen Datensatz und `C` (column) enthält den Wert.

17.6 Logeinträge filtern

- **Mehrfach vorkommende Werte einer Spalte ausblenden**
  Mit `getAvailable` blenden Sie Datensätze aus, die in einer ausgewählten Spalte mehrfach vorkommen (entspricht `select distinct`):

```
<Request id="3" type="getAvailable" restrict="is_moduleName"/>
```

Das Ergebnis sind alle unterschiedlichen Werte der angegebenen Spalte, z. B.:

```
<Response size="3" id="4">
    <Column name="is_moduleName" like="">
      <Available>CIM-CSV-Parser(8677)</Available>
      <Available>backupAll(2399098)</Available>
      <Available>retryFtp(2398355)</Available>
    </Column>
</Response>
```

- **Daten filtern**
  Die Ergebniswerte können Sie in einer weiteren Abfrage als Filter verwenden, Filter können auch das Schlüsselwort „LIKE“ enthalten:

```
<Request type="getData">
   <Column name="is_moduleName" like="CIM" />
</Request>
```

  Das Ergebnis sind alle Datensätze der Tabelle, bei denen die Werte in der Spalte „is_moduleName“ die Zeichenkette CIM enthalten.
- **Min- und Max-Filter**
  Für Spalten mit einem numerischen Datentyp (z. B. long, int, float) und Datumsangaben (z. B. java.util.Date) können die Filter „min“ und „max“ gesetzt werden, z. B.:

```
<Request id="3" type="getData">
   <Column name="is_workflowId" min="30"
max="245"/>
</Request>
```

  Es werden alle Datensätze mit einer WorkflowID größer/gleich 30 bzw. kleiner/gleich 245 angezeigt.
- **Tabellenspalten ausblenden**
  Mit `setColumns` definieren Sie eine Tabellen-View:

```
<Request id="1" type="setColumns">
   <Column name="is_workflowId"/>
   <Column name="is_status"/>
   <Column name="is_priority">
</Request>
```

  Es werden nur die Tabellenspalten ausgegeben, die im `Request`-Element gelistet sind.

17.7 Logeinträge sortieren

Zum Sortieren von Datensätzen verwenden Sie die `sort`- und `sortLevel`-Attribute wie folgt:

- `sort="[asc|desc]"` gibt an, ob die Werte in der Spalte aufsteigend oder absteigend sortiert werden sollen

- `sortLevel` legt die Priorität bei Sortieren fest.

```
<Request type="getData">
  <!-- sortLevel=0: zuerst wird anhand des Namens sortiert -->
    <Column name="is_workflowName"sort="asc" sortLevel="0"/>
  <!-- sortLevel=1: dann anhand der ID -->
    <Column name="is_workflowID"sort="desc"
       sortLevel="1"/>
</Request>
```

17.8 In Logeinträgen nach Zeichenketten suchen

Wenn Ihre Log-Tabelle Spalten mit dem Attribut `like` besitzt, können Sie diese nach (Teil)Zeichenketten durchsuchen lassen:

```
<Request id="3" type="getData">
    <Column name="is_moduleType" like="IS Conf"/>
</Request>
```

Zur Suche nach Teilen von Zeichenketten sind keine Wildcards nötig. Die Suche nach einem Modul „IS“ würde auch ein Modul mit dem Namen „IS Config“ finden.

17.9 Größe von Eingangsnachrichten ermitteln

Sie können die Größe der Eingangsnachrichten ermitteln lassen, indem Sie dem inubit IS Configuration Utility eine Moduleigenschaft hinzufügen. Die Größe wird in der Workflow-Variablen `DataStreamSize` ausgegeben.

So gehen Sie vor

1. Fügen Sie den Moduleigenschaften des inubit IS Configuration Utilities die Eigenschaft `Operation` mit dem Datentyp `Text (einzeilig)` hinzu.
   → Siehe *Moduleigenschaften hinzufügen (Workbench: Benutzer-Guide, Kap. 3.7, S. 123)*.
   In der Eigenschaftentabelle werden der Name `Operation` und ein Eingabefeld angezeigt.

2. Geben Sie den Wert `getDataStreamSize` ein.
3. Speichern Sie Ihre Angaben.

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Line Reader Eigenschaften“, S. 229*

Verwendung

Mit dem Line Reader können Sie zeilenweise Text aus einer text-basierten Eingangsnachricht, z. B. einer HTML- oder XML-Datei, in die Ausgangsnachricht kopieren.

18.1 Dialog „Line Reader Eigenschaften“

In diesem Dialog haben Sie folgende Optionen:

Grundeinstellungen

- **Zeichensatzkodierung**
  Geben Sie die Zeichensatzkodierung der Eingangsnachrichten ein oder wählen Sie einen Eintrag aus der Liste.
- **Start Index**
  Geben Sie an, ab welcher Zeile Text kopiert werden soll. Die Zeilennummerierung beginnt bei 0.
- **Blockgröße**
  Geben Sie an, wie viele Zeilen ab dem Startindex kopiert werden sollen.
  Um alle Zeilen ab dem Startindex zu kopieren, geben Sie „-1“ ein.
- **Zeilenumbrüche**
  Um die Zeilenumbrüche aus den Eingangsnachrichten beizubehalten, markieren Sie die Option „Übernehmen“.
  Um die Zeilenumbrüche der Eingangsnachrichten zu konvertieren, wenn z. B. für die Weiterverarbeitung andere Zeilenumbrüche nötig sind, wählen Sie den gewünschten Zieltyp (Unix oder Windows) aus der Liste aus.

Dieser Abschnitt erläutert die folgenden Themen:

- *Eingangsnachricht für PDF Toolbox erzeugen, S. 231*
- *Beispiel-Workflow, S. 232*

Verwendung

Sie nutzen die PDF Toolbox, um innerhalb eines Workflows mehrere PDF-Dateien in einer PDF-Datei zusammenzuführen.

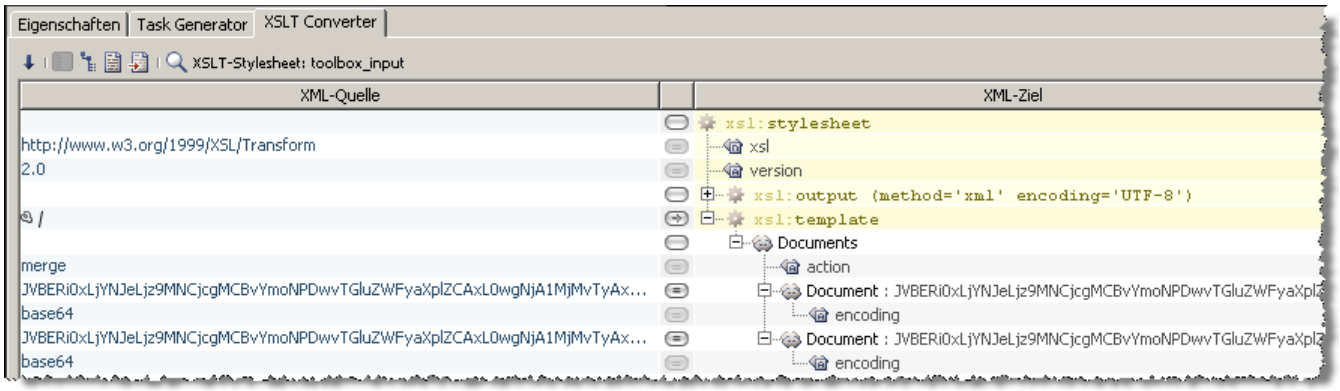
Das PDF Toolbox Modul hat keine konfigurierbaren Eigenschaften, erwartet jedoch eine Eingangsnachricht in einem bestimmten XML-Format.

Wie die PDF-Dateien zusammengeführt werden, bestimmen Sie durch den Workflow.

19.1 Eingangsnachricht für PDF Toolbox erzeugen

Das PDF Toolbox Modul erwartet eine Eingangsnachricht in einem bestimmten XML-Format, die Sie mit einem XSLT Converter erzeugen.

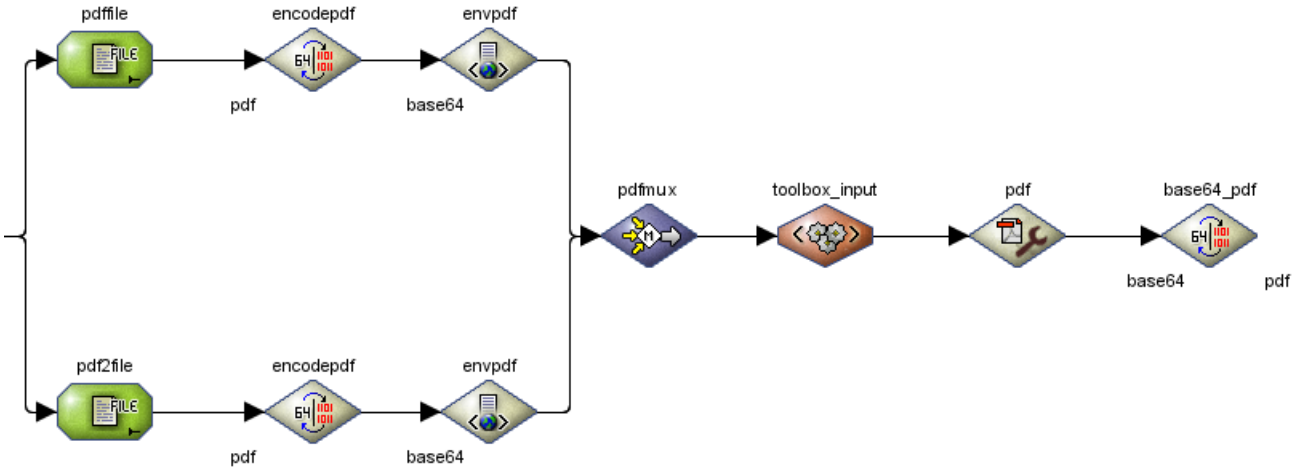
Für ein korrektes Mapping muss das XSLT-Stylesheet folgendermaßen aussehen:



→ Siehe auch

- *XSLT Converter (Data Converter) (Workbench/Process Engine: Modul-Guide, Kap. 5, S. 133)*
- XSLT Converter konfigurieren: *Technical Workflow erstellen 3: CSV-XML konvertieren nach openTRANS (Tutorials, Kap. 6.5, S. 116)*

19.2 Beispiel-Workflow



Der abgebildete Workflow erzeugt aus zwei PDF-Dateien eine PDF-Datei.

PDF > Base64

Die Encoder erzeugen aus den eingelesenen PDF-Dateien jeweils base64-kodierte Ausgangsnachrichten, um aus den binären Eingangsdaten ein Übergangsdateiformat zur XML-basierten Weiterverarbeitung zu generieren.

Wählen Sie im Dialog folgende Einstellung:

- Einstellungen: „Daten kodieren“ > Algorithmus wählen: base64“

→ Siehe *Encoder (Utility) (Workbench/Process Engine: Modul-Guide, Kap. 16, S. 215).*

Base64 > XML

Die XML Enveloper erzeugen aus den base64-kodierten Eingangsnachrichten XML-formatierte Ausgangsnachrichten.

→ Siehe *XML Enveloper (Utility) (Workbench/Process Engine: Modul-Guide, Kap. 24, S. 249).*

XML > XML

Der Multiplexer führt mehrere eingehende XML-Nachrichten in eine einzelne XML-Nachricht zusammen.

→ Siehe *Multiplexer (Workbench/Process Engine: Modul-Guide, Kap. 28.8, S. 268).*

XSLT Converter

Diese XML-Ausgangsnachricht gibt der Multiplexer an einen XSLT Converter weiter, der daraus die Eingangsnachricht für die PDF Toolbox erstellt.

→ Siehe *Eingangsnachricht für PDF Toolbox erzeugen (Workbench/ Process Engine: Modul-Guide, Kap. 19.1, S. 231).*

XML > Base64

Der XSLT Converter leitet die XML-Ausgangsnachricht an die PDF Toolbox weiter, die eine einzelne base64-kodierte PDF-Datei erzeugt.

Base64 > PDF

Die base64-kodierte Ausgangsnachricht der PDF Toolbox wird zur Dekodierung an ein Encoder/Decoder Modul weitergegeben. Beim Dekodieren werden die Daten wieder in ihr ursprüngliches PDF-Format zurückgeführt.

Dieser Abschnitt erläutert die folgenden Themen:

- *Process Data Logger konfigurieren , S. 235*
- *Dialog „Process Data Logger Eigenschaften“, S. 236*

Verwendung

Mit dem „Process Data Logger“ können Sie prozess-spezifische Daten protokollieren und in eine Datenbank schreiben lassen. Sie können z. B.

- auf aktuelle Werte von Systemvariablen zugreifen,
- Werte aus XML-basierten Eingangsnachrichten protokollieren,
- eine Datenbankspalte mit statischen Werten füllen und das Variablen-Mapping nutzen, um die Werte bei der Ausführung des Workflows mit Workflow-spezifischen Werten zu überschreiben.

inubit AG empfiehlt, mindestens die globale Prozess-ID, die Workflow-Prozess-ID und Zeitstempel zu protokollieren.

Die protokollierten Daten werden im Register „Monitoring > Process Log“ angezeigt.

→ Siehe *Fachliche Prozessdaten im Process Log anzeigen (Workbench: Benutzer-Guide, Kap. 13.11, S. 353).*

20.1 Process Data Logger konfigurieren

Voraussetzungen

Sie benötigen eine Beispiel-Nachricht, mit derselben Struktur wie die zu dokumentierenden XML-Daten.

So gehen Sie vor

1. Erzeugen Sie ein Modul vom Typ „Process Data Logger“.
2. Wählen Sie im *Dialog „Process Data Logger Eigenschaften“ (Workbench/Process Engine: Modul-Guide, Kap. 20.2, S. 236)* das Logging-Ziel aus, in dem die zu protokollierenden Prozessdaten gespeichert werden soll. Das Logging-Ziel kann z. B. die IS Log-Datenbank sein.
3. Führen Sie eine der folgenden Aktionen aus, um anzugeben, in welche Tabelle die Daten gespeichert werden sollen:
   - Wählen Sie eine vorhandene Tabelle aus.

- Um eine neue Tabelle zu erzeugen, klicken Sie auf den Button „DB Manager“. Der *Dialog „Datenbankmanager“ (Process Engine: Administrator- und Entwickler-Guide, Kap. 8.4, S. 117)* öffnet sich.

In der Abbildungstabelle werden alle Spalten der ausgewählten Tabelle angezeigt.

4. Geben Sie für jede Spalte an, aus welcher Quelle die zu protokollierenden Daten stammen.
   - **Spalte mit Elementwert aus XML-Eingangsnachricht füllen**
     a. Markieren Sie in der entsprechenden Zeile die Option „D“.
     b. Um das XML-Element festzulegen, dessen Daten protokolliert werden sollen, doppelklicken Sie in derselben Zeile in das Feld „Wert“. Der -Button wird eingeblendet.
     c. Klicken Sie auf . Der XPath-Assistent öffnet sich.
     d. Laden Sie Ihre Beispiel-XML-Nachricht.
     e. Ziehen Sie das gewünschte XML-Element in das Feld „XPath“.
     f. Schließen Sie den Dialog mit „OK“.
   - **Spalte mit Wert einer Systemvariablen füllen**
     a. Markieren Sie in der entsprechenden Zeile die Option „V“.
     b. Klicken Sie in das Feld „Wert“ und wählen Sie aus der Liste eine Systemvariable aus.
   - **Spalte mit statischem Wert füllen**
     a. Markieren Sie keine Option (weder „D“ noch „V“).
     b. Doppelklicken Sie in das Feld „Wert“ und geben Sie eine Zeichenkette ein. Um längere Texte komfortabel einzugeben, klicken Sie auf . Ein Texteditor wird angezeigt.
   - **Spalte nicht füllen**
     Markieren Sie rechts neben der Spalte die Option „Ignorieren“. Damit wird die Spalte beim Befüllen nicht beachtet.
5. Wenn Sie mit dem Konfigurieren aller Datenbankspalten in der Abbildungstabelle fertig sind, klicken Sie auf „Fertig stellen“.

Der Dialog schließt sich. Sie können nun den konfigurierten Process Data Logger verwenden.

20.2 Dialog „Process Data Logger Eigenschaften“

In diesem Dialog haben Sie folgende Optionen:

Logging-Ziel

- **Datenbank**
  Ziel, in dem die zu protokollierenden Daten gespeichert werden. Es werden alle bereits publizierten und für den aktuellen Benutzer sichtbaren Datenbank-Konnektoren angezeigt.
  Ein weiteres Ziel ist die Logdatenbank der inubit Process Engine „IS Log Database“.
- **Tabelle/DB-Manager**
  Zur Auswahl einer Tabelle in der angegebenen Datenbank.
  Falls noch keine geeignete Tabelle vorhanden ist oder eine vorhandene Tabellenstruktur geändert werden soll, öffnet der Button „DB Manager“ den *Dialog „Datenbankmanager“ (Process Engine: Administrator- und Entwickler-Guide, Kap. 8.4, S. 117)*“ zum Erstellen einer oder mehrerer Tabellen.

Abbildungstabelle

Die Tabelle zeigt alle Spalten der ausgewählten Tabelle an. Es wird für jede Tabellenspalte festgelegt, ob diese mit Daten gefüllt werden soll und wenn ja, aus welcher Quelle die zu protokollierenden Daten stammen.

- **XPath des Bezugsknotens**
  Element in der Eingangsnachricht mit folgenden Funktionen:
  - Pfade zu Elementen, die unterhalb des Bezugsknoten liegen, können relativ zum Bezugsknoten angegeben werden.
  - Wenn das Element in der Eingangsnachricht mehrfach vorhanden ist, wird darüber iteriert.
- **Tabellenspalten**
  - **Spalte D** (Daten):
    Wenn markiert, dann wird die dazugehörige Spalte der Datenbanktabelle mit Werten aus XML-formatierten Eingangsnachrichten gefüllt.
    Im Feld „Wert“ kann ein Editor zum Laden der Beispielnachricht geöffnet werden.
  - **Spalte V** (Variable):
    Wenn markiert, dann wird die entsprechende Spalte der Datenbanktabelle mit dem Wert einer Systemvariablen gefüllt.
    Im Feld „Wert“ wird eine Liste aller Systemvariablen zur Auswahl angeboten.

Wenn keine der beiden Optionen markiert ist, dann kann als Wert eine beliebige Zeichenkette eingegeben werden.

  - **Spalte „Wert“**
    Mögliche Angaben sind
    - XPath eines XML-Elements,
    - beliebige Zeichenkette,
    - Systemvariable.

- **Spalte „Ignorieren“**
  Wenn markiert, dann wird die angegebene Tabellenspalte ignoriert und nicht befüllt.
  Wenn die Option nicht markiert ist und kein Wert zum Befüllen angegeben ist, dann wird die Spalte mit “ “ gefüllt.

- **Zurücksetzen** (Button)
  Lädt die zuletzt gespeicherten Moduleinstellungen noch einmal. Entspricht einem Klick auf „Abbrechen“ und erneutem Öffnen des Dialogs.

→ Siehe *Process Data Logger konfigurieren (Workbench/Process Engine: Modul-Guide, Kap. 20.1, S. 235).*

Dieser Abschnitt erläutert die folgenden Themen:

- *Einsatz unter UNIX/Linux, S. 239*
- *Prozessmodell-Report erstellen, S. 240*
- *Prozessmodell-Report anpassen, S. 240*
- *Dialogbeschreibungen, S. 242*

Verwendung

Mit dem Report Data Collector können Sie mit wenig Aufwand eine umfassende Dokumentation Ihres Prozessmodells erzeugen.

Der Report Data Collector erzeugt Reports mit Screenshots der verwendeten technischen und fachlichen Diagramme, Module und Formulare. Sie können Inhalt und Layout der Reports entsprechend Ihren Wünschen anpassen.

21.1 Einsatz unter UNIX/Linux

Standardmäßig sind die Applikationsserver Tomcat und JBoss so konfiguriert, dass der Einsatz des Report Data Collectors keinen laufenden X-Server voraussetzt. Allerdings ist ohne grafische Oberfläche nur ein Teil der Funktionen des Report Data Collectors nutzbar. So können z. B. nur Screenshots von Workflows erzeugt werden, jedoch keine Screenshots von Formularen.

Voraussetzungen

Um die volle Funktionalität des Report Data Collectors nutzen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Der X-Server ist gestartet.
- Aus dem Startskript des Applikationsservers wurde die Option `JVM_PARAMS="$JVM_PARAMS -Djava.awt.headless=true"` entfernt und danach wurde der Applikationsserver neu gestartet.
  Das Startskript finden Sie unter:
  - **JBoss**: `<iS-installdir>server/JBoss/bin/run.conf.bat` bzw. `run.conf`
  - **Tomcat**: `<iS-installdir>server/Tomcat/bin/setenv.bat` bzw. `setenv.sh`

21.2 Prozessmodell-Report erstellen

Sie können Ihr Prozessmodell mit wenigen Klicks erzeugen lassen und dabei festlegen, ob Ihr gesamtes Prozessmodell dokumentiert werden soll oder Teile davon und welche Informationen über die Prozesse enthalten sein sollen.

Zum Erzeugen des Prozessmodells wird ein mitgelieferter und vorkonfigurierter Workflow genutzt, in dem u. a. das Layout und das Logo des Reports definiert sind.

→ Siehe *Prozessmodell-Report anpassen (Workbench/Process Engine: Modul-Guide, Kap. 21.3, S. 240).*

So gehen Sie vor

1. Wählen Sie in der inubit Workbench in der Menüleiste „Datei > Prozessmodell-Report erstellen“. Der Report-Assistent öffnet sich.

Die Option wird nur angezeigt, wenn Sie das Modul „Process Data Logger“ lizenziert haben!

2. Lassen Sie sich von dem Assistenten durch die Erstellung des Reports führen.

   → Siehe auch

   - *Dialog „Diagrammauswahl“ (Workbench/Process Engine: Modul-Guide, Kap. 21.4.1, S. 242)*
   - *Dialog „Konfiguration der Auswahl“ (Workbench/Process Engine: Modul-Guide, Kap. 21.4.2, S. 242)*.

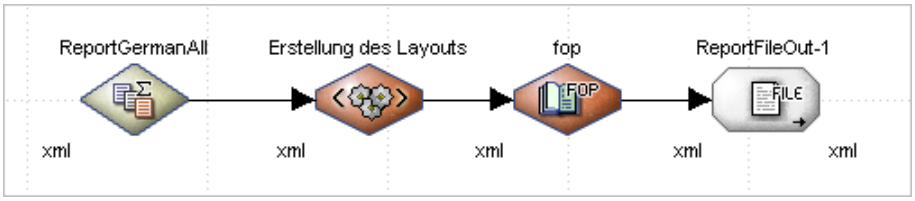
Nach Beenden des Report-Assistenten wird der Report als PDF-Datei erzeugt und sofort anzeigt.

21.3 Prozessmodell-Report anpassen

Zum Erzeugen des Prozessmodell-Reports wird ein vorkonfigurierter Workflow genutzt, in dem u. a. das Layout und das Logo des Reports definiert sind. Sie können diesen Workflow beliebig anpassen und ihn z. B. mit einem Workflow Starter erweitern, um die Reportgenerierung zu automatisieren.

Überblick

Der System-Workflow „Report-Prozessmodell“ sammelt Prozessdaten, erzeugt einen PDF-Report und speichert dieses in ein Exportverzeichnis:



1. **Report Data Collector**
   Stellt die Daten für den Report als XML-Datei zusammen. Abbildungen und Screenshots werden im Verzeichnis `<iS-installdir>\server\ibis_root\<BenutzernameOderGruppe>\workflow\report prozessmodell` abgelegt.
2. **XSLT Converter**:
   Erzeugt das Layout für den Report.
3. **FO Converter**:
   Erzeugt eine PDF-Datei namens `report.pdf`.
4. **File Connector**
   Speichert die PDF-Datei in das Verzeichnis `<iS-installdir>\server\ibis_root\<BenutzernameOderGruppe>\workflow\report prozessmodell\export`.

So gehen Sie vor

1. Importieren Sie den System-Workflow `Report-Prozessmodell` aus dem Verzeichnis `<iS-installdir>/client/templates/workflows/Report_Prozessmodell.zip` in die inubit Workbench.
2. Passen Sie den Workflow an:
   - Die Auswahl der Diagramme legen Sie im Report Data Collector fest.
   - Um den Report an das Corporate Design Ihrer Firma anzupassen, können Sie das Logo der Reports austauschen:
     a. Erstellen Sie eine GIF-Datei mit Ihrem Logo. Das Logo muss 450 Pixel breit sein und 161 Pixel hoch.
        Größere Logos werden proportional an die fest definierte Größe angepasst!
     b. Speichern Sie die Datei als `is_logo.gif` in das Verzeichnis `<iS-installdir>\server\ibis_root\conf\images`.
3. Führen Sie den Workflow aus.

21.4 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Diagrammauswahl“, S. 242*
- *Dialog „Konfiguration der Auswahl“, S. 242*

21.4.1 Dialog „Diagrammauswahl“

Verfügbare Diagramme

Die Baumstruktur enthält alle Diagramme, die in den Report aufgenommen werden können.

Markieren Sie ein oder mehrere Diagramme oder Diagrammgruppen und klicken Sie auf den „>“ Button, um diese in den Report aufzunehmen.

Alle Diagramme auf der inubit Process Engine

Nur sichtbar, wenn Sie als Benutzer `root` angemeldet sind.

Um alle fachlichen und technischen Workflows aller Gruppen und Benutzer in den Report aufzunehmen.

Ausgewählte Diagramme

Zeigt alle ausgewählten Diagramme an.

Mit dem „<“-Button entfernen Sie Diagramme aus der Auswahl.

21.4.2 Dialog „Konfiguration der Auswahl“

- **Sprache des Reports**
  Sprache der vorkonfigurierten Report-Elemente, z. B. „Inhaltsverzeichnis/Table of Contents“, „erzeugt am/generated on“.
- **Sprache der Diagramme**
  Sprache der Diagramm- und Elementkommentare und der sonstigen Elementbezeichner.
  Wenn Sie „Default“ wählen, dann wird die Sprache ausgegeben, die im Register „Default“ eingegeben wurde.
  → Siehe *Mehrsprachige Texte eingeben (Workbench: Benutzer-Guide, Kap. 2.19.3, S. 97).*
- **Dokumententitel**

Bezeichnung Ihres Prozessmodellreports. Wird auf der Titelseite angezeigt.

Business Process Diagramme, Systemdiagramme, Organigramme

Zum Festlegen, wie Diagramme in den Report aufgenommen werden:

- **Abbildung des Diagramms**
  Wenn markiert, dann wird von jedem ausgewählten Diagramm ein Screenshot erstellt und dem Report zugefügt.
- **Verlinkte Diagramme einbeziehen**
  Wenn markiert, dann werden alle Diagramme in den Report einbezogen, die mit einem ausgewählten Diagramm über Linkpoints verbunden sind.
  Auf diese Weise können Sie Ihr vollständiges Prozessmodell in den Report aufnehmen lassen, vorausgesetzt, alle Diagramme auf den fachlichen und technischen Ebenen sind verlinkt.
- **Elementliste (einschl. der Eigenschaften)**
  Wenn markiert, dann wird für jedes Diagramm zusätzlich eine Liste aller enthaltenen Elemente und deren Eigenschaften erstellt.

Technische Workflows, BPEL Diagramme

- **Abbildung des Diagramms**
  Wenn markiert, dann von jedem Diagramm ein Screenshot erstellt und dem Report zugefügt.
- **Modulliste (einschl. der Eigenschaften)**
  Wenn markiert, dann wird für jedes Diagramm zusätzlich eine Liste aller enthaltenen Elemente und deren Eigenschaften erstellt.
- **Modul-spezifische Eigenschaften**
  Wenn markiert, dann enthält der Report zusätzlich Abbildungen mit Informationen für folgende Module:
  - **EDI Format Adapter**: Konfigurierte Regeln
  - **Flat File Format Adapter**: Konfigurierte Regeln
  - **XSLT Converter**: XSLT Converter Tabelle
  - **Web Services Connector**: WSDL-Datei
  - **Task Generator**: Formulare

Einleitende Beschreibung

(Optional)

Einleitungstext für den Prozessmodellreport.

Dieser Abschnitt erläutert die folgenden Themen:

- *Schlüssel und Zertifikate erstellen, S. 245*
- *Dialogbeschreibungen, S. 246*

Verwendung

Mit dem S/MIME Modul können Sie Eingangsnachrichten im MIME-Format verschlüsseln, entschlüsseln, signieren sowie vorhandene Signaturen prüfen.

22.1 Schlüssel und Zertifikate erstellen

Für die Verschlüsselung benötigen Sie Schlüssel, diese können Sie mit dem „keytool“ von Java erstellen. Den Algorithmus, der zur Schlüsselerzeugung eingesetzt werden soll, geben Sie beim Erzeugen der Schlüssel an.

→ Für Informationen über den Aufruf von keytool siehe *SSL-Verbindung und Server-Authentifizierung konfigurieren (Process Engine: Administrator- und Entwickler-Guide, Kap. 5.1, S. 55)*.

Beispielaufruf „Schlüssel erzeugen“

```
keytool
-genkey
-dname "O=inubit AG, C=de"
-validity 365
-keystore smime.keystore
-keypass inubit
-storepass inubit
-alias inubit
-keyalg RSA
```

Beispielaufruf „Zertifikat erzeugen“

```
keytool
-export
-keystore smime.keystore
-alias inubit
-file smime.cer
```

22.2 Dialogbeschreibungen

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „S/MIME-Konfiguration“, S. 246*

22.2.1 Dialog „S/MIME-Konfiguration“

- **MIME verschlüsseln**
  Ausgehende Nachrichten werden im S/MIME-Format verschlüsselt.
  Dafür müssen Sie als Sender den öffentlichen Schlüssel des Empfängers kennen. Diese Information hinterlegen Sie im folgenden Dialog.
- **MIME entschlüsseln**
  Das Utility soll eingehende Nachrichten entschlüsseln. Dafür benötigen Sie den privaten Schlüssel des Empfängers. Diese Information hinterlegen Sie im nächsten Dialog.
- **MIME signieren**
  Das Utility signiert die ausgehenden Nachrichten, das heißt, die Nachrichten werden mit einer verschlüsselten Prüfsumme versehen.
  Dafür benötigen Sie einen privaten Schlüssel, diese Information hinterlegen Sie im nächsten Dialog. Der Empfänger der Nachrichten muss den dazugehörigen öffentlichen Schlüssel besitzen, um die Korrektheit der Signatur prüfen zu können.
- **MIME-Signatur prüfen**
  Überprüft die Korrektheit der Signaturen in der Eingangsnachricht, indem die Signaturen mit dem öffentlichen Schlüssel des Absenders verglichen werden. Diese Information hinterlegen Sie im nächsten Dialog.

Dieser Abschnitt erläutert die folgenden Themen:

- *Ausgangsnachrichten, S. 247*
- *Dialog „XML Diff Moduleinstellungen“, S. 248*

Verwendung

Das XML Diff-Utility ermittelt die Unterschiede zwischen XML-Dokumenten, XML-Elementen oder Variablen, die XML-Dokumente enthalten.

Das Ermitteln der Differenzen wird häufig auf einer älteren und einer neueren Version eines XML-Dokuments ausgeführt, um zwischenzeitlich erfolgte Änderungen zu ermitteln.

Das Ermitteln der Differenzen von XML-Dokumenten ist sehr rechenintensiv, daher empfiehlt inubit AG eine leistungsstarke CPU mit hoher Taktrate und mehreren Cores.

23.1 Ausgangsnachrichten

Die Differenzen werden in einer XML-Ausgangsnachricht in Insert-, Update- und Delete-Elementen dargestellt und beziehen sich jeweils auf die Textknoten von Elementen und Attributen, z. B.:

```
<?xml version="1.0" encoding="UTF-8"?>
<XmlDifferences count="3">
  <Difference>
    <Type>delete</Type>
    <Path>
      /ns1:Root/Document/Position[3]/Price
    </Path>
    <nodeType>Element</nodeType>
    <nodeName>Price</nodeName>
    <oldValue>12.36</oldValue>
  </Difference>
  <Difference>
    <Type>update</Type>
    <Path>
      /ns1:Root/Document/Position[3]/Article
    </Path>
    <nodeType>Element</nodeType>
    <nodeName>Article</nodeName>
    <oldValue>table</oldValue>
```

```
    <newValue>chair</newValue>
  </Difference>
  <Difference>
    <Type>insert</Type>
    <Path>/ns1:Root/@orderID</Path>
    <nodeType>Attribute</nodeType>
    <nodeName>orderID</nodeName>
    <newValue>0815</newValue>
  </Difference>
</XmlDifferences>
```

23.2 Dialog „XML Diff Moduleinstellungen“

In diesem Dialog legen Sie fest, welche XML-Strukturen (XML-Dokumente, -Elemente oder Variablen) miteinander verglichen werden sollen.

Basisdokument

Markieren Sie eine der Optionen:

- **D**: Zur Auswahl eines Elements in einem XML-Dokument.
- **V**: Zur Angabe einer Workflow-Variablen vom Typ `xs:anyType`, die ein XML-Dokument enthält.
- **Fest**: Zum Import eines statischen XML-Dokuments in das Modul.
- **Quelle**: Wählen Sie mit dem Button „Basisdokument setzen“ das XML-Dokument aus, das als Vergleichsquelle dient.

Vergleichsdokument

Siehe „Basisdokument“.

Objekt-Identifikation

Bei einer Liste von Objekten mit Identifikatoren können Sie die Verarbeitung des Vergleichs beschleunigen, indem Sie das Diff zerlegen und einzelne Objekte der Vergleichsdokumente einzeln in den Diff verschieben. Dazu geben Sie in der Spalte „Objekt-Iterator“ den Pfad zu dem Element an, über das iteriert werden soll, und wählen in der Spalte „Objekt-Identifikator“ das zugehörige Identifikator-Element oder -Attribut.

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „XML Enveloper Eigenschaften“, S. 249*

Verwendung

Ein XML Enveloper erzeugt eine XML-formatierte Ausgangsnachricht aus text-basierten Eingangsnachrichten.

Daten, die nicht der XML-Syntax entsprechen, müssen vor Benutzung des XML Enveloper umgewandelt werden! Binärdaten können Sie z. B. mit dem base64-Verfahren kodieren.

24.1 Dialog „XML Enveloper Eigenschaften“

Einstellungen

- **Zeichensatz der Quelldaten/Zeichensatz des Zieldokuments**
  Standard ist der Zeichensatz ISO-8859-1. Dies ist der Standardzeichensatz der alle Zeichen der gebräuchlichen westeuropäischen Sprachen enthält.
  Darüber hinaus sind die Zeichensätze US-ASCII, UTF-8, UTF-16BE, UTF-16LE und UTF-16 wählbar.
  Wenn nötig, können Sie jeden anderen beliebigen Zeichensatz eintragen.
- **Name des XML-Root-Elements**
  Der Name „DATA“ ist voreingestellt. Sie können einen beliebigen, XML-konformen Namen wählen.
- **CDATA Element** (Sonderzeichen ignorieren):
  Wählen Sie diese Option, wenn die eingehende Nachricht reservierte XML-Sonderzeichen enthält, z. B. „<“, „>“ oder „/“. Wenn die Option markiert ist, dann wird der gesamte Inhalt der Eingabedatei in ein CDATA-Element eingeschlossen (beginnend mit dem Tag „`<![CDATA[`“ und endend mit „`]]>`“). CDATA Abschnitte werden bei der Syntaxanalyse ignoriert.
  Beispiel: **Eingabe-Textdatei**:
  „Die Zahl 9 ist größer als die Zahl 7, dargestellt als: 9 > 7. Für die Division wird auch das Zeichen / benutzt. In XML werden Tags in spitze Klammern eingeschlossen. `<ITEM>` ist ein Beispiel.“
  Ausgabe-XML-Datei:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DATA>
<![CDATA[Die Zahl 9 ist größer als die Zahl 7,
```

```
 dargestellt als: 9 > 7
 Für die Division wird auch das
 Zeichen / benutzt.
 In XML werden Tags in spitze Klammern
 eingeschlossen.
 <ITEM> ist ein Beispiel.]]>
</DATA>
```

Dieser Abschnitt erläutert die folgenden Themen:

- *Beispiel für Ein- und Ausgangsnachrichten, S. 251*
- *Dialog „XML Formatter Eigenschaften“, S. 252*

Verwendung

Mit dem XML Formatter können Sie Eingangsnachrichten im XML-Format formatieren lassen.

Da XML kein Layout vorschreibt, kann der Inhalt einer XML-Datei z. B. in einer einzigen langen Zeile stehen. Diese Darstellung ist vom Menschen bei umfangreichen Dateien nur schlecht lesbar. Der XML Formatter fügt Einrückungen und Zeilenumbrüche in eine beliebige XML-Datei und erzeugt damit eine gut lesbare Struktur.

25.1 Beispiel für Ein- und Ausgangsnachrichten

Eingangsnachricht:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Invoice>
  <test1>
    <ArticleId>1</ArticleId>
    <Price>1</Price>
  </test1>
</Invoice>
```

Ausgangsnachricht

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Invoice>
  <test1>
    <ArticleId>1</ArticleId>
    <Price>1</Price>
  </test1>
</Invoice>
```

25.2 Dialog „XML Formatter Eigenschaften“

Einrückungen

- **XML-Elemente einrücken**
  Mit dieser Option werden die XML Elemente entsprechend ihrer hierarchischen Position eingerückt.
- **Anzahl der Leerzeichen**
  Bestimmt, wieweit ein Element jeweils eingerückt wird.
  Um nach Änderungen des Wertes den vorgegebenen Wert wiederherzustellen, klicken Sie auf „Standard“.

Zeilenumbrüche

Zeilenumbrüche hinzufügen

Fügt nach 80 Zeichen einen Zeilenumbruch ein. Dies verhindert horizontales Scrollen, in einigen Fällen mindert diese Option jedoch die Übersichtlichkeit.

Leerzeichen

Führende und nachfolgende Leerzeichen entfernen

Entfernt Leerzeichen vor und hinter dem Wert eines Elements.

Namensräume

Aktiviert die folgenden Optionen zum Optimieren der Namensräume:

- **Namensräume optimieren**
  Entfernt alle doppelten und ungenutzten Namensraum-Deklarationen.
- **Folgende Namensräume entfernen**
  Entfernt alle in einer Liste genannten Namensraum-Deklarationen; zum Erstellen der Liste klicken Sie auf „Liste bearbeiten“.
- **Alle Namensräume entfernen**
  Entfernt alle Namensraum-Deklarationen.

Kodierung

- **Zeichensatz der Ausgangsnachricht**
  Zur Auswahl der Zeichensatzkodierung.
  Standard ist der Zeichensatz ISO-8859-1. Dies ist der Standardzeichensatz der alle Zeichen der gebräuchlichen westeuropäischen Sprachen enthält (inkl. der länderspezifischen Sonderzeichen).
  Alternativ können Sie jeden anderen beliebigen Zeichensatz eingeben.

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „XML Validator Eigenschaften“, S. 253*

Verwendung

Ein XML Validator überprüft bei beliebigen XML-Eingangsnachrichten, ob diese

- wohlgeformt sind
- oder einem internen/externen XML Schema
- bzw. einer internen/externen DTD entsprechen.

Abhängig von der Größe des XML Schemas bzw. der DTD und dem Ort, an dem diese Datei erreichbar ist, kann das XML Validator Modul die Performance der inubit Process Engine negativ beeinflussen. Um die Einbußen zu minimieren, sollten Sie die DTD bzw. das Schema im lokalen Dateisystem speichern statt diese über einen Webserver oder eine externe Adresse (z. B. HTTP oder FTP) verfügbar zu machen

Ergebnis des Validators

Abhängig vom Ergebnis der Validierung werden Eingangsnachrichten folgendermaßen weitergereicht:

- Validierung war erfolgreich
  Die Eingangsnachricht wird unverändert als Ausgangsnachricht weitergeleitet.
- Validierung schlug fehlt
  Die Eingangsnachricht wird in folgender Form weitergeleitet:

```
<Validation>
        <Errors>
            <Error [Fehlermeldung]</Error>
        <Errors>
        <OriginalDocument>
            [Inhalt derEingangsnachricht]
        </OriginalDocument>
</Validation>
```

26.1 Dialog „XML Validator Eigenschaften“

In diesem Dialog haben Sie folgende Optionen:

Überprüfung der Eingangsnachricht

- **Nur prüfen, ob Eingangsnachricht wohlgeformtes XML enthält**
  Wenn markiert, dann wird lediglich geprüft, ob die Eingangsnachrichten den XML-Regeln entsprechen, u. a.:
  - Jede Nachricht besitzt genau ein Wurzelelement.
  - Jedes Element mit Inhalt hat ein Beginn- und ein End-Tag.
  - Die Beginn- und End-Tags sind ebenentreu-paarig verschachtelt.
  - Kein Element hat mehrere Attribute mit demselben Namen.

  Wenn die Prüfung fehlschlägt, wird der Fehlerausgang genutzt.
- **XML Schema-Prüfung der Eingangsnachricht**
  Wenn markiert, dann werden die Eingangsnachrichten gegen die XML Schemas geprüft, welche in den Eingangsnachrichten enthalten oder referenziert sind.
- **DTD-Prüfung der Eingangsnachricht**
  Wenn markiert, dann werden Eingangsnachrichten gegen die DTD geprüft, welche in den Eingangsnachrichten enthalten oder referenziert sind.

Externes XML Schema oder externe DTD verwenden

Um Eingangsnachrichten gegen ein XML Schema bzw. eine DTD prüfen zu lassen, welche über einen URI referenziert ist oder statisch in das Modul geladen wird:

- **URI**
  Der Zugriff kann über das http-, ftp- oder file-Protokoll erfolgen, z. B.
  - `file:///usr/local/schema/test.xsd`
  - `http://www.inubit.com/schema/test.xsd`
  - `file:///C:/example.xsd`
- **Statische Datei**
  Nach dem Laden werden der Dateiname, die Dateigröße und der MIME-Typ der Datei im Dialog angezeigt.
- **XML Schema/DTD cachen**
  Die Dateien werden vorkompiliert und im Arbeitsspeicher gehalten, um die Performance des Moduls zu verbessern.

Weitere Einstellungen

- **Validierung nach Fund des ersten Fehlers sofort abbrechen**
  Die Validierung stoppt beim ersten Fehler statt die gesamte Eingangsnachricht zu prüfen.
- **Bei fehlerhafter Validierung Fehlerausgang verwenden**
  Wenn die Validierung fehlschlägt und im Workflow ein Fehlerausgang vorhanden ist, dann wird dieser Fehlerausgang aktiviert.

Dieser Abschnitt erläutert die folgenden Themen:

- *Invoke, S. 255*
- *OnMessage, S. 257*
- *Receive, S. 257*
- *Reply, S. 259*

Verwendung

Web Service Controls sind in BPEL-Prozessen die zentralen Elemente zum Aufrufen und Anbieten von Web Services.

In Technical Workflows können Web Service Controls als Alternative zu Web Services Connectoren eingesetzt werden.

→ Siehe *Web Services mit BPEL erstellen (Tutorials, Kap. 10, S. 159).*

27.1 Invoke

Ein Invoke-Control ruft genau eine Operation eines Web Services auf und nimmt dessen Rückgabenachricht (wenn vorhanden) entgegen.

Das Control hat keine spezifischen Eigenschaften, sondern wird im Workflow konfiguriert.

Legen Sie einen PartnerLink mit einem Partner-Endpunkt an, bevor Sie das Control konfigurieren!

27.1.1 Dialog beschreibung

Service-Definition

- **PartnerLink**
  Zur Auswahl des aufzurufenden PartnerLinks.
- **Adresse des aufzurufenden Service**
  Wird automatisch angezeigt, sobald ein PartnerLink ausgewählt ist.

Aufzurufende Operation

- **Operation**
  Zur Auswahl der Operation, die das Invoke-Control aufrufen soll.

In Technical Workflows: Standardmäßig ist die Operation nicht festgelegt, und es wird versucht, die Operation anhand der übergebenen SOAP-Nachricht zu ermitteln.

- **Eigenschaften anhand übergebener SOAP-Nachricht ermitteln**
  Die Option ist nur aktiviert, wenn Sie keine Operation festgelegt haben.
- **SOAP-Action**
  Geben Sie eine SOAP-Action an, wenn sie nicht aus der SOAP-Nachricht ermittelt werden kann oder soll.
- **Die aufzurufende Aktion liefert kein Ergebnis zurück.**
  Aktivieren Sie die Checkbox, wenn die aufzurufende Aktion kein Ergebnis zurückliefert.

Eingabe/Rückgabe-Daten

- **Eingabe/Ausgabenachricht verwenden**
  Nur in Technical Workflows.
  Wenn markiert, dann wird der Web Service mit der Eingangsnachricht des Controls aufgerufen und die Rückgabenachricht (wenn vorhanden) wird wieder an das Control übergeben. Analog zu einem Web Services Output Connector.
  - **Mit kompletter SOAP-Nachricht**
    Wenn markiert, dann wird als Eingabe eine komplette SOAP-Nachricht mit Envelope und Body benötigt bzw. als Ausgabe eine komplette SOAP-Nachricht ausgegeben.
  - **Mit nur Nutzdaten (Inhalt von SOAP-Body)**
    Wenn markiert, dann wird als Eingabe nur das Element innerhalb des SOAP-Body benötigt bzw. in der Ausgangsnachricht nur der Inhalt des SOAP-Body ausgegeben.
- **Variablen verwenden**
  Wenn markiert, dann wird der Web Service Request aus der gewählten Eingabe-Variable erzeugt. Falls der Web Service eine Response zurück gibt, dann wird diese in die Rückgabe-Variable geschrieben.
  - **Eingabe-Variable**
    Zur Auswahl einer vorhandenen Variablen bzw. zum Anlegen einer neuen Variablen.
  - **Rückgabe-Variable**
    Zur Auswahl einer vorhandenen Variablen bzw. zum Anlegen einer neuen Variablen.

27.2 OnMessage

Verwendung

Ein OnMessage-Control führt dieselben Aktionen wie ein Receive-Modul durch, wird jedoch immer in Verbindung mit einem Pick-Modul eingesetzt.

Dieses Control hat keine spezifischen Eigenschaften, sondern wird im Workflow konfiguriert.

Legen Sie einen PartnerLink mit einem lokalen Endpunkt an, bevor Sie das Control konfigurieren!

→ Siehe
- *Receive (Workbench/Process Engine: Modul-Guide, Kap. 27.3, S. 257)*
- *Pick: Verwendung (Workbench: Benutzer-Guide, Kap. 11.6.3, S. 289)*

27.3 Receive

Dieser Abschnitt erläutert die folgenden Themen:

■ *Dialog „Web Service Einstellungen“, S. 258*

Verwendung

- Ein Receive-Control wartet am Anfang eines Workflow auf einen Web Service Request. Sobald der Request eintrifft, werden die Daten aus dem Request-Body entpackt und in die angegebene Variablen gefüllt. Der Workflow wird gestartet.
- Innerhalb eines Workflows wartet das Control auf Callbacks eines asynchronen Web Services.

Dieses Control hat keine spezifischen Eigenschaften, sondern wird im Workflow konfiguriert.

Legen Sie einen PartnerLink mit einem lokalen Endpunkt an, bevor Sie das Control konfigurieren!

27.3.1 Dialog „Web Service Einstellungen“

In diesem Dialog haben Sie folgende Optionen:

Modus

Zur Auswahl der Funktionalität des Moduls:

- **Request entgegennehmen und Workflow starten**
  Modul arbeitet als Listener und Workflow-Starter.
- **Asynchronen Callback entgegennehmen und Workflow fortführen**
  Modul arbeitet als Callback-Listener, insbesondere beim Aufruf asynchroner Web Services.

Service-Definition

- **PartnerLink**
  Zur Auswahl des PartnerLinks.
- **Adresse des bereitgestellten Service**
  Wird automatisch angezeigt, sobald der PartnerLink ausgewählt ist.

Akzeptierte Operation

- **Operation**
  Zur Auswahl der Operation, die das Control anbietet.

Empfangene Daten

- **Ausgangsnachricht setzen**
  Nur in Technical Workflows.
  Wenn markiert, dann übergibt das Control den empfangenen Request an den Workflow (vergleichbar mit Web Services Input Connectoren).
  - **Mit kompletter SOAP-Nachricht**
    Wenn markiert, dann wird eine komplette SOAP-Nachricht ausgegeben.
  - **Nutzdaten (Inhalt von SOAP-Body)**
    Wenn markiert, dann wird nur der Inhalt des SOAP-Body ausgegeben.
- **Variablen-Inhalt setzen**
  Wenn markiert, dann schreibt das Control den empfangenen Request in die angegebene Variable.
  - **Variable**
    Zum Auswählen einer bereits deklarierten Variable bzw. zum Anlegen einer neuen Variablen.

27.4 Reply

Ein Reply-Control gibt das Ergebnis eines Technical Workflows oder BPEL-Prozesses als SOAP-Response an den aufrufenden Web Service zurück.

27.4.1 Dialogbeschreibung

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Reply-Moduleigenschaften“ in Technical Workflows, S. 259*
- *Dialog „Reply-Moduleigenschaften“ in BPEL-Diagrammen, S. 260*

27.4.1.1 Dialog „Reply-Moduleigenschaften“ in Technical Workflows

Dieser Dialog bietet folgende Optionen:

Antwortdaten

- **Eingangsnachricht als Ausgabe verwenden**
  Sinnvoll z. B. in Kombination mit einem Web Services Input Connector, um die Rückgabe von Werten explizit darzustellen und damit die Lesbarkeit des Workflows zu verbessern.
- **Ausgabe aus Variableninhalt erzeugen**
  Erzeugt aus einer Eingangsnachricht eine SOAP-Nachricht, die als Response verwendet wird.

Antwort-Erzeugung aus Variableninhalt

- **Modul, das eingehende Anfrage entgegennimmt**
  Angabe, welches Receive- bzw. onMessage-Modul oder welcher Web Services Input Connector den Request entgegengenommen hat, zu dem die Response erstellt werden soll.
  - **Operation**
    Zeigt an, welche Operation das Modul anbietet.

Art der Antwort

- **Normale Antwort**
  Erzeugt eine positive Response, entsprechend dem Rückgabeparameter, der in der Web-Service Definition definiert ist.
  - **Variable**

Name der Variablen, aus deren Inhalt die Antwort generiert wird.

- **Fehler-Antwort**
  Nur aktiv, wenn in der Web-Service Definition ein Fehler-Rückgabeparameter definiert ist.
  - **Fehler-Name/Fehler-Variable**
    Liste der definierten Fehler-Namen und Fehler-Variablen.

27.4.1.2 Dialog „Reply-Moduleigenschaften“ in BPEL-Diagrammen

Dieser Dialog bietet folgende Optionen:

Antwort-Erzeugung aus Variableninhalt

- **Modul, das eingehende Anfrage entgegennimmt**
  Auswahl des Receive- bzw. onMessage-Moduls, welches den Request entgegengenommen hat, zu dem die Response erstellt werden soll.
  - **Operation**
    Zeigt an, welche Operationen das Modul anbietet.

Art der Antwort

- **Normale Antwort**
  Erzeugt eine positive Response, entsprechend dem Rückgabeparameter, der im PartnerLink definiert ist.
  - **Variable**
    Name der Variablen, aus deren Inhalt die Antwort generiert wird.
- **Fehler-Antwort**
  Nur aktiv, wenn in der Web Service-Definition ein Fehler-Rückgabeparameter definiert ist.
  - **Fehler-Name/Fehler-Variable**
    Liste der definierten Fehler-Namen und Fehler-Variablen.

Dieser Abschnitt erläutert die folgenden Themen:

- *Assign, S. 261*
- *Compensate, S. 262*
- *Compensate Scope, S. 262*
- *Demultiplexer, S. 263*
- *Empty, S. 267*
- *Exit, S. 267*
- *Joiner, S. 268*
- *Multiplexer, S. 268*
- *Rethrow, S. 272*
- *Splitter, S. 272*
- *Throw, S. 273*
- *Validate, S. 274*
- *Wait, S. 274*
- *Workflow Connector, S. 275*
- *Workflow Starter, S. 278*

Verwendung

Mit den Workflow Controls der inubit Suite 6 bilden Sie die Prozesslogik ab, um in Technical Workflows und BPEL-Diagrammen den Ablauf der Ausführung zu steuern.

Mit den Workflow Controls können Sie alle gängigen EAI und Workflow Pattern implementieren.

28.1 Assign

Verwendung

An einem Assign-Modul modifizieren Sie Variablenwerte. Sie können z. B.

- Variablenwerte kopieren
- Ein-/Ausgangsnachrichten auf Variablen abbilden
- Variablen validieren (Unterstützung für validate-Attribut)

Dieser Modultyp hat keine konfigurierbaren Eigenschaften. Die Variablenwerte modifizieren Sie im Workflow mit Hilfe des Variablen-Mappings.

→ Siehe

- *BPEL-Diagramme (Workbench: Benutzer-Guide, Kap. 11, S. 283)*
- *Workflow-Variablen und Mappings (Workbench: Benutzer-Guide, Kap. 14, S. 365)*

28.2 Compensate

Verwendung

Ein Compensate-Modul wird immer mit dem Fehler- oder Kompensationsausgang eines Scopes verbunden, der mindestens einen weiteren Scope enthält.

Sobald der konfigurierte Fehler- oder Kompensationsfall auftritt, aktiviert das Compensate-Module alle Kompensationsausgänge von bereits erfolgreich ausgeführten inneren Scopes. Damit werden die Aktionen der inneren Scopes rückgängig gemacht. Dies ist vergleichbar mit dem Rollback-Mechanismus bei Datenbank-Transaktionen.

Dieser Modultyp hat keine konfigurierbaren Eigenschaften.

→ Siehe *BPEL-Diagramme (Workbench: Benutzer-Guide, Kap. 11, S. 283).*

28.3 Compensate Scope

Verwendung

Die Funktion dieses Moduls ist vergleichbar mit dem Compensate-Modul. Anders als beim Compensate-Modul wird aber lediglich der Kompensationsausgang eines einzigen, definierten Scopes aktiviert.

→ Siehe *Kompensation in einem Scope konfigurieren (Workbench: Benutzer-Guide, Kap. 12.4.2, S. 306).*

Dieser Modultyp hat keine konfigurierbaren Eigenschaften. Der aufzurufende Scope bzw. dessen Kompensationsausgang wird an dem Modul im Workflow definiert.

28.4 Demultiplexer

Dieser Abschnitt erläutert die folgenden Themen:

- *Bedingungen für die Nachrichtenweiterleitung definieren, S. 263*
- *Dialog „Demultiplexer Konfiguration“, S. 265*

Verwendung

Der Demultiplexer leitet eine Eingangsnachricht in einen von mehreren Ausgängen, vorausgesetzt, die Bedingungen sind erfüllt, die an dem Ausgang definiert sind.

Zusätzlich kann das Modul XML-basierte Eingangsnachrichten vor der Weiterleitung in mehrere Teile splitten.

28.4.1 Bedingungen für die Nachrichtenweiterleitung definieren

Die Bedingungen für die Weiterleitung der Eingangsnachricht legen Sie fest, nachdem Sie den Demultiplexer in ein Diagramm eingefügt und mit den nachfolgenden Modulen verbunden haben.

So gehen Sie vor

1. Führen Sie eine der folgenden Aktionen aus:
   - Doppelklicken Sie eine Verbindung zwischen Demultiplexer und nachfolgendem Modul.
   - Markieren Sie eine Verbindung, öffnen Sie das Kontextmenü und wählen Sie „Demultiplexer Konfiguration ändern“.

   Der *Dialog „Demultiplexer Konfiguration“ (Workbench/Process Engine: Modul-Guide, Kap. 28.4.2, S. 265)* wird angezeigt.
2. Legen Sie fest, unter welchen Umständen die gewählte Verbindung genutzt werden soll:
   - Immer, wenn keine Bedingung oder Business Rule zutrifft:

     Markieren Sie die Option „Diesen Ausgang benutzen, wenn keine Bedingungen an anderen Verbindungen zutreffen“.
   - Wenn eine oder mehrere Bedingungen zutreffen:

     Klicken Sie auf den Button „Bedingung hinzufügen“.

     Die Eingabefelder zum Definieren von Bedingungen werden eingeblendet.

     Eine Bedingung besteht aus folgenden Elementen:
     - Wert in der XML-Eingangsnachricht selbst oder ein Variablenwert in der Nachricht oder ein XPath-Ausdruck, der sich auf die XML-Eingangsnachricht bezieht.

- Ein Vergleichswert (XML-Daten, Variablen oder einer Zeichenkette)
- Ein Operator, mit dem Sie die Art des Vergleichs festgelegen.

Sie können für jede Verbindung mehrere Bedingungen anlegen und diese mit UND oder ODER verknüpfen.

- Wenn eine Business Rule zutrifft:

  Klicken Sie auf den Button „Business Rules verwenden“. Die Bedienfelder zum Auswählen einer Routing-Regel werden eingeblendet.

  Die Routing-Regel muss bereits angelegt und im Repository gespeichert worden sein.

  → Siehe *Business Rules einsetzen (Workbench: Benutzer-Guide, Kap. 15, S. 397).*

3. **XML-Eingangsnachricht teilen**

   Eine XML-Eingangsnachricht kann in mehrere Ausgangsnachrichten aufgeteilt werden. Damit können Sie z. B. eine XML-Eingangsnachricht, die mehrere Bestellungen in `order`-Elementen enthält, in mehrere Nachrichten mit jeweils einer Bestellung in einem `order`-Element aufteilen. Wenn solch eine Nachricht z. B. an den `order`-Elementen aufgeteilt wird, dann ist `order` das Trennelement und wird in den neu erzeugten Teilnachrichten zum root-Element. Das Trennelement wird mit einem XPath-Ausdruck definiert.

   a. Um den XPath-Ausdruck zu erzeugen, klicken Sie auf den Button neben der Option „XML-Splitting-Element“. Der XPath-Assistent öffnet sich.

      → Siehe *XPath-Assistent (Workbench: Benutzer-Guide, Kap. 1.16, S. 67).*

   b. Laden Sie eine Beispielnachricht, die das Trennelement enthält, und ziehen Sie das Trennelement in das XPath-Feld.

   c. Schließen Sie den Dialog mit „OK“.

4. Um die nächste Verbindung zu konfigurieren, wählen Sie im Bereich „Modul für Verbindung“ das nächste Modul aus, dessen Verbindung Sie mit Bedingungen belegen möchten, und wiederholen die Schritte 2 und 3.
5. Klicken Sie auf „OK“, um die konfigurierten Verbindungen zu speichern und den Dialog zu schließen.

28.4.2 Dialog „Demultiplexer Konfiguration“

Modul für Verbindungen

Liste aller Module, die mit dem Tool verbunden sind. Im Dialog wird immer die Konfiguration des aktuell markierten Moduls bzw. dessen Ausgang angezeigt.

Die Ausgänge werden in der Reihenfolge abgearbeitet, in der sie in der Liste angezeigt werden. Die Abarbeitungsreihenfolge wird mit den Pfeil-Buttons über der Liste geändert.

Beschreibung

Erläuternder Kommentar. Wird auf der Verbindungslinie zum markierten Modul angezeigt.

Bedingungen

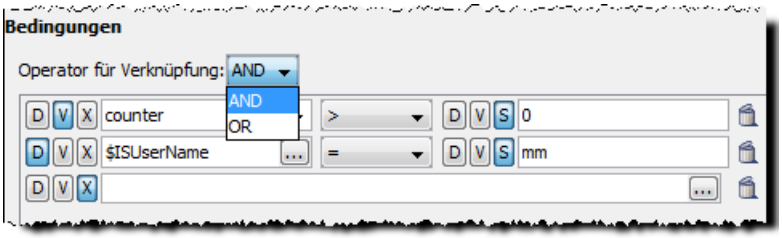
- **Diesen Ausgang benutzen, wenn keine Bedingungen an anderen Verbindungen zutreffen**

  Legt die Verbindung als Standard-Ausgang fest für alle Nachrichten, die keine der anderen Bedingungen erfüllen. Die Verbindung wird automatisch mit „otherwise“ benannt.

  Wenn die Option nicht markiert ist und eine Nachricht gar keine Bedingung erfüllt, dann wird ein Fehler erzeugt und im Monitoring angezeigt.

- **Bedingung hinzufügen** (Button)

  Blendet die folgenden Bedienfelder zum Festlegen von Bedingungen ein:



  - **Operator für Verknüpfungen**: Zum Verknüpfen von mehreren Bedingungen.
    - **AND**: Alle Bedingungen müssen zutreffen.
    - **OR**: Eine der Bedingungen muss zutreffen.
  - **Button „D“**: Die Bedingung bezieht sich auf ein Element in der XML-basierten Nachricht. Im Eingabefeld muss ein XPath-Ausdruck eingegeben werden.

    Falls eine Beispiel-XML-Datei vorliegt, klicken Sie auf den ...-Button und wählen das Element aus.
  - **Button „V“**: Die Bedingung bezieht sich auf eine Variable. Alle vorhandenen Variablen werden in der nebenstehenden Liste angezeigt.

- **Button „X“**: Die Bedingung bezieht sich auf einen XPath-Ausdruck. Der XPath wird auf einem leeren Dokument ausgeführt (nicht auf dem Datenstrom). Der ...-Button öffnet einen Assistenten zum Anlegen des XPath-Ausdrucks.
- **Operator**
  Zur Auswahl eines Operators für den Werteabgleich. Es gibt folgende Operatoren:

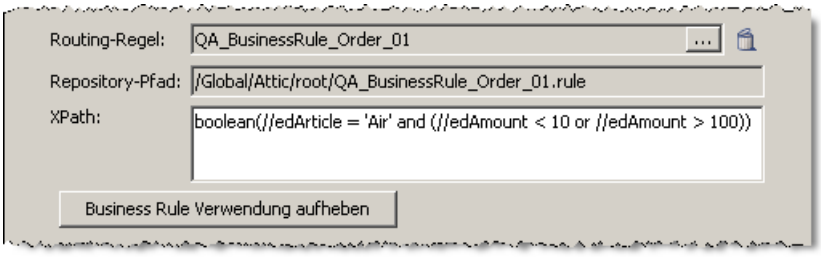
Operator	Bedeutung
=	Gleich
<	Kleiner als
>	Größer als
>=	Größer gleich
<=	Kleiner gleich
!=	Ungleich
Exists	Existiert
NotExists	Existiert nicht
XPath	XPath-Ausdruck

- **Buttons „D“ und „V“: ö**Siehe oben.
- **Button „S“:** Die Bedingung bezieht sich auf die eingegebene Zeichenkette.

Beispiele:

- `/AUFTRAG/AUFTRAGSSUMME > 1000`
- `/AUFTRAG/LIEFERUNG/ = Express`
- `/IBISProfile/Profile/Name Exists`
  Wenn der Knoten `Name` in der Eingangsnachricht vorhanden ist, dann wird in diesen Ausgang verzweigt.
- `count(/Order/test1)` > 1
  Wenn die Auswertung der XPath-Funktion `count()` ergibt, dass der Knoten `/Order/test1` mehr als einmal existiert, dann wird in diesen Ausgang verzweigt.

- **Business Rules verwenden** (Button)
  Blendet die folgenden Bedienfelder zum Zuordnen einer Business Rule ein:



→ Siehe *Regeln auf technischer Ebene zuordnen (Workbench: Benutzer-Guide, Kap. 15.7, S. 406).*

Splitting Konfiguration

- **XML-Splitting-Element**
  XML-basierte Eingangsnachrichten können an einem ausgewählten Element aufgeteilt werden, sodass z. B. aus einer Rechnung mit vier Einzelposten vier Rechnungen mit je einem Posten entstehen. Das Splitting-Element wird durch einen XPath-Ausdruck definiert und wird in den neu erzeugten Teilnachrichten zum `root`-Element.
  Der Button neben der Option öffnet den XPath-Assistenten zum Definieren des XPath-Ausdrucks.
  → Siehe *XPath-Assistent (Workbench: Benutzer-Guide, Kap. 1.16, S. 67).*
- **Splitting auf übergeordneter Ebene durchführen**
  Erweitert das Splitting um den Elternknoten des Splitting-Elements. Diese Option ist nützlich, wenn der Elternknoten Informationen enthält, die in jeder Nachricht gebraucht werden, die durch das Splitting entsteht.

28.5 Empty

Verwendung

Ein Empty-Modul führt keine Aktionen durch. Es übergibt die Eingangsnachrichten ohne Änderungen an das nachfolgende Modul.

Ein Empty-Modul kann z. B. als einziges Modul in einem Fehlerausgang verwendet werden, falls ein Fehler zwar abgefangen, aber nicht weiter behandelt werden soll.

Dieser Modultyp hat keine konfigurierbaren Eigenschaften.

28.6 Exit

Verwendung

Das Exit-Modul beendet die Ausführung eines Workflows.

Es kann z. B. an Verzweigungen eingesetzt werden, um die Workflow-Ausführung zu beenden, wenn keine Bedingungen an den verschiedenen Verbindungen zutreffen.

Dieser Modultyp hat keine konfigurierbaren Eigenschaften.

28.7 Joiner

Verwendung

Ein Joiner verbindet zwei oder mehr eingehende Nachrichten und leitet diese als eine Ausgangsnachricht an das nachfolgende Modul weiter. Die Nachrichten werden dabei nicht verändert.

Dieser Modultyp hat keine konfigurierbaren Eigenschaften.

28.8 Multiplexer

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Multiplexer Eigenschaften“, S. 269*

Verwendung

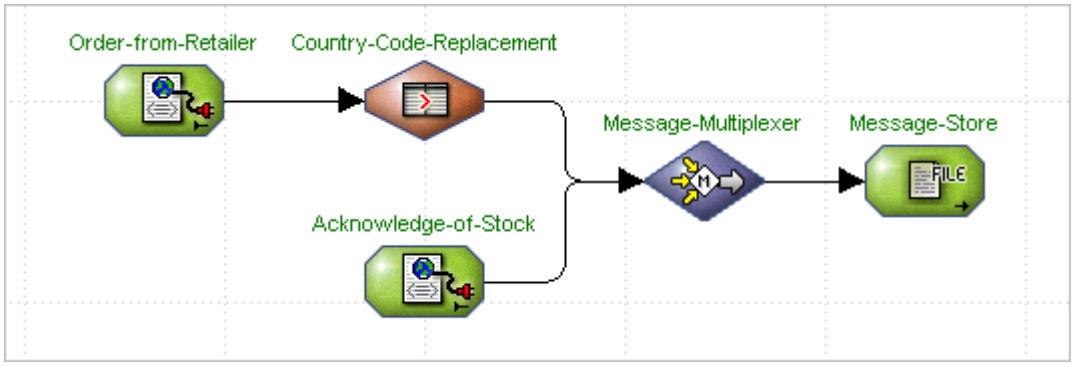
Ein Multiplexer führt mehrere XML-Eingangsnachrichten, die zu verschiedenen Zeitpunkten eintreffen, zu einer XML-Nachricht zusammen.

Zusammenführung von Variablen

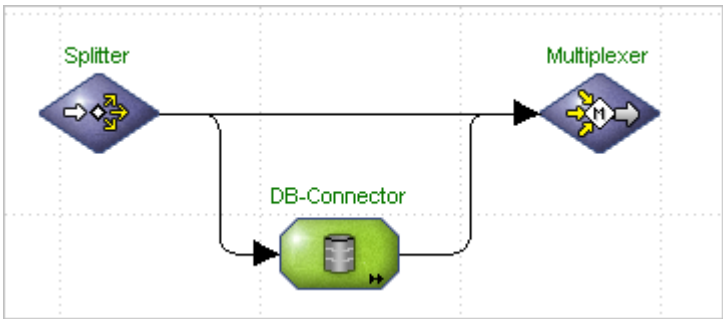
Die Variablen aller eingehenden Zweige werden zusammengeführt. Die Basis bilden dabei die Variablen des letzten durchlaufenen Zweigs. Die Variablen der anderen Zweige werden immer dann hinzugefügt, wenn deren Variablenname in der Basis noch nicht existiert.

Beispiel

Die folgende Abbildung zeigt, wie zwei Nachrichten aus zwei Konnektoren am Multiplexer zu einer Nachricht zusammengeführt und weitergeleitet werden:



In der folgenden Abbildung fügt der Multiplexer eine Nachricht aus dem Splitter mit den Ergebnissen einer Datenbankabfrage zusammen:



Konfiguration

Der Modultyp hat keine konfigurierbaren Eigenschaften.

Wie die Eingangsnachrichten zusammen geführt werden sollen, legen Sie im Workflow an einer der Eingangsverbindungen fest.

Format der Ausgangsnachricht

Die Ausgangsnachricht hat ein iS-spezifisches XML-Format, das weitgehend konfigurierbar ist.

28.8.1 Dialog „Multiplexer Eigenschaften“

Aufruf

Im Technical Workflow > Kontextmenü des Multiplexer-Eingangs > „Multiplexer Konfiguration ändern“

In diesem Dialog legen Sie fest, wie die Eingangsnachrichten des Multiplexers zusammengeführt werden.

Die Einstellungen für das Envelope-XML-Element und den Modus gelten für alle eingehenden Verbindungen des Multiplexers. Alle anderen Einstellungen gelten für jeweils die Verbindung, die in der Liste „Modul für Verbindung“ markiert ist.

XML Root Envelope

Envelope-XML-Element für ausgehende Nachricht

Name des Wurzel-Elements der zu erzeugenden XML-Ausgangsnachricht.

Modus

- **Nur Nachrichten von einer WF-Ausführung verwenden**

Wenn aktiviert, dann verarbeitet der Multiplexer nur Nachrichten, die zu derselben Workflow-Ausführung gehören und daher dieselbe Prozess-ID (Systemvariable `ISProcessID`) haben.

- **Zeitgesteuerte Verarbeitung der Nachricht (FIFO)**
  Wenn aktiviert ist, dann werden die Nachrichten in der Reihenfolge ihres Eintreffens am Multiplexer verarbeitet („first in, first out“).

Im FIFO-Modus werden Nachrichten verschiedener Workflow-Ausführungen nicht getrennt voneinander behandelt!

Modul

Modul für Verbindung

In der Liste werden alle Module angezeigt, die mit dem Eingang des Multiplexers verbunden sind.

Um eine Verbindung zu konfigurieren, markieren Sie diese in der Liste.

Sie müssen jede Verbindung einzeln konfigurieren. Wenn nicht alle Verbindungen konfiguriert sind, dann werden die Nachrichten zeitgesteuert verarbeitet (FIFO).

Beschreibung

Beschreibung der Verbindung. Die Beschreibung wird auf der Verbindungslinie angezeigt.

Wertabgleich

- **XML-Element aus Eingangsnachricht für Wertabgleich nutzen**
  Wenn aktiviert ist, dann wird in allen Eingangsnachrichten der Wert eines ausgewählten XML-Elements geprüft und es werden immer nur die Nachrichten zusammengeführt, deren Werte identisch sind.
  Mit dieser Funktion können Sie z. B. alle Nachrichten mit derselben Auftragsnummer zusammenführen lassen.
- **XML-Element**
  Angabe eines XPath zu dem XML-Element für den Wertabgleich. Alternativ lassen Sie den XPath erzeugen, indem Sie über den -Button eine Beispieldatei laden und das Element in der Datei markieren.

Allgemein

- **Envelope-XML-Element für Nachricht setzen**
  Wenn markiert, dann wird die Eingangsnachricht in ein Element mit dem von Ihnen angegebenen Namen eingeschlossen.
- **XML-Root-Element in Eingangsnachricht ignorieren**

Wenn die Eingangsnachricht ohne ihr Wurzelelement mit den anderen Nachrichten zusammengeführt werden soll, wählen Sie „Ja“, andernfalls „Nein“.

- **Eingangsnachricht erforderlich**

  Wenn Sie „Ja“ wählen, wartet der Multiplexer auf die Eingangsnachricht. Die Ausführung des Workflows wird erst fortgesetzt, wenn die Nachricht eingetroffen ist.

  Wenn Sie „Nein“ wählen, verarbeitet der Multiplexer die Eingangsnachricht, wenn diese vorliegt. Falls aber die Eingangsnachricht noch nicht vorhanden ist, während an allen anderen Verbindungen die Bedingungen zum Fortsetzen des Workflows bereits erfüllt sind, dann wartet der Multiplexer bis zum Ablauf des Timeouts. Danach wird der Workflow fortgesetzt.

- **Eingangsnachricht ignorieren**

  Wenn markiert, dann wird das Eintreffen der Eingangsnachricht als Signal genutzt, um die Fortsetzung des Workflows auszulösen (vorausgesetzt, die Bedingungen an den anderen Verbindungen sind erfüllt).

  Die Eingangsnachricht selbst wird nicht weitergleitet.

- **Timeout**

  Legt fest, wie lange der Multiplexer an diesem Eingang auf das Eintreffen der Eingangsnachricht warten soll.

  Wenn die angegebene Zeitdauer überschritten und die Eingangsnachricht nicht eingetroffen ist, dann

  - wird der Workflow mit einem Fehler abgebrochen, wenn die Eingangsnachricht erforderlich war,
  - wird der Workflow fortgesetzt, wenn die Nachricht nicht erforderlich war.

  Das Timeout beginnt, wenn die erste Nachricht am Multiplexer eintrifft. Ab dann werden auch die Timeouts für alle weiteren, noch nicht eingetroffenen Nachrichten berechnet.

  Ein Timeout =0 bedeutet, dass der Multiplexer unendlich lange wartet.

- **Anzahl der Eingangsnachrichten**

  Erwartete Anzahl von Nachrichten pro Workflow-Ausführung.

28.9 Rethrow

Verwendung

Ein Rethrow-Modul wird in verschachtelten Scopes verwendet, um einen Fehler, der vom aktuellen Fehlerausgang nicht behandelt wird, an den Fehlerausgang des nächsten, äußeren Scope weiterzuleiten.

Anders als bei Einsatz eines Throw-Moduls wird bei einem Rethrow-Modul immer dieselbe Fehler-Instanz weitergeleitet, während bei einem Throw-Modul ein neuer Fehler ausgelöst wird.

28.10 Splitter

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Splitter“, S. 272*

Verwendung

Ein Splitter vervielfacht die Eingangsnachrichten und leitet sie an die nachfolgenden Module weiter. Die Anzahl der Ausgangsnachrichten entspricht der Anzahl der ausgehenden Verbindungen des Splitters.

Dieser Modultyp hat keine konfigurierbaren Eigenschaften. Die Abarbeitungsreihenfolge der Verbindungen legen Sie im Workflow fest.

28.10.1 Dialog „Splitter“

Modul für Verbindungen

In der Liste werden alle Module angezeigt, die mit dem Splitter verbunden sind.

Die Verbindungen werden in der Reihenfolge abgearbeitet, in der sie in der Liste angezeigt werden. Die Abarbeitungsreihenfolge wird mit den Pfeil-Buttons über der Liste geändert.

28.11 Throw

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Throw-Einstellungen“, S. 273*

Verwendung

Mit einem Throw-Control können Sie Fehler selbst definieren und z. B. durch den Fehlerausgang eines Scopes abfangen und behandeln. Den Namen des Fehlers und die Variable, in welche die Informationen über den Fehler geschrieben werden sollen, legen Sie nach dem Einfügen des Throw in den Workflow fest.

Bei der Ausführung eines Throw wird standardmäßig ein Eintrag im Queue Manager erzeugt und eine Fehlermail versendet. Dieses Verhalten können Sie abschalten.

Beim Testen wird der Fehler im Standardfehlerdialog angezeigt.

Dieses Control hat keine konfigurierbaren Moduleigenschaften.

28.11.1 Dialog „Throw-Einstellungen“

Aufruf

Durch einen Klick auf den „Einstellungen“-Button nach dem Einfügen des Throw-Controls in den Workflow:

Optionen

- **Fehlername**
  Name des Fehlers. Der Fehlername wird genutzt, um den Fehler zum passenden Fehlerausgang am Scope zuzuordnen.
- **Fehlervariable**
  Auswahl der Variable, in welche die Fehler-Detaildaten geschrieben werden.
- **Eintrag im Queue Manager erzeugen und Error-Mail verschicken**
  Wenn markiert, verursacht der Fehler einen Eintrag im Queue Manager, zusätzlich wird eine E-Mail versendet.
  → Für Informationen, wie Sie den Empfänger der E-Mail konfigurieren siehe *Bei fehlerhafter Ausführung Nachricht an (Workbench: Benutzer-Guide, Kap. 2, S. 107)*.

28.12 Validate

Dieser Abschnitt erläutert die folgenden Themen:

■ *Dialog „Validate-Moduleinstellungen“, S. 274*

Verwendung

Das Validate-Control prüft ausgewählte Variablenwerte gegen XML Schemas oder WSDL-Dateien, die am Workflow referenziert sind.

Wenn die Validierung erfolgreich war, dann läuft der Prozess weiter. Andernfalls wird ein Fehler geworfen und das Validierungsergebnis als Teil der Fehlermeldung ausgegeben.

→ Siehe *Fehlerbehandlung und -unterdrückung (Workbench: Benutzer-Guide, Kap. 12.3, S. 299).*

28.12.1 Dialog „Validate-Moduleinstellungen“

In diesem Dialog wählen Sie die Variablen aus, die validiert werden sollen.

Die Liste enthält alle Variablen, die im Workflow definiert sind. Ein Klick auf den Plus-Button übernimmt die angezeigte Variable in den Anzeigebereich über der Liste.

28.13 Wait

Verwendung

Ein Wait-Modul hält die Ausführung eines Workflows für eine bestimmte Dauer, bis zum Erreichen eines bestimmten Zeitpunkts oder entsprechend dem Ergebnis des XPath-Ausdrucks an.

Der Workflow wechselt beim Warten vom Status „Processing“ in den Status „Waiting“.

Die Einstellungen des Moduls legen Sie im Workflow über das Kontextmenü des Moduls fest.

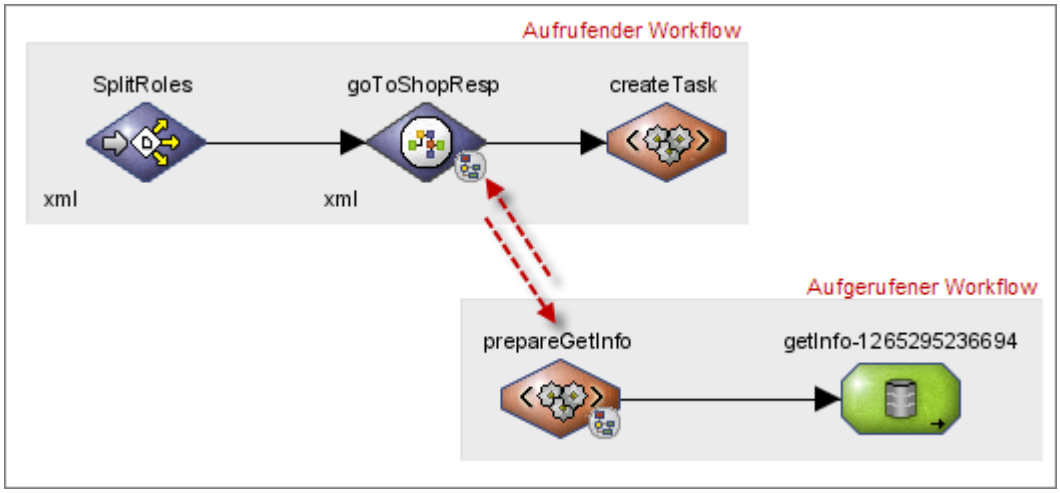
28.14 Workflow Connector

Dieser Abschnitt erläutert die folgenden Themen:

■ *Dialogbeschreibungen „Workflow Connector“, S. 276*

Verwendung

Ein Workflow Connector verbindet einen Technical Workflow mit einem Subworkflow und ermöglicht damit die Mehrfachverwendung von Technical Workflows in anderen Technical Workflows.



Beim Anlegen des Workflow Connectors können Sie u. a. festlegen, ob der Subworkflow synchron oder asynchron aufgerufen werden soll:

- Synchroner Aufruf: Der aufrufende Workflow wartet auf die Ergebnisse des Subworkflows. Der Workflow Connector gibt das Ergebnis des letzten Moduls im Subworkflow zurück. Falls ein Fehler auftritt, hängt die Verarbeitung von der Konfiguration der Fehlerbearbeitung ab.
- Asynchroner Aufruf: Der aufrufende Workflow wird nach dem Aufruf direkt fortgesetzt.

Die Übergabe von Workflow-Variablen konfigurieren Sie, nachdem Sie den Workflow Connector in ein Diagramm eingefügt und mit dem vorhergehenden Modul verlinkt haben.

Sie können einen Workflow Connector manuell in einen Technical Workflows einfügen und mit einem anderen Technical Workflow verlinken oder bereits bestehende Technical Workflows nachträglich modularisieren, dabei werden die Workflow Connectoren automatisch eingefügt.

→ Siehe

- *Variablenübergabe bei verlinkten Technical Workflows konfigurieren (Workbench: Benutzer-Guide, Kap. 12.7, S. 312)*
- *Refactoring: Technical Workflows mit Workflow Connectoren verbinden (Workbench: Benutzer-Guide, Kap. 12.6, S. 311)*

- In verlinkten Technical Workflows gibt es zusätzliche Systemvariablen siehe *Systemvariablen-Übersicht (Workbench: Benutzer-Guide, Kap. 14.6, S. 373)*.

28.14.1 Dialogbeschreibungen „Workflow Connector“

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Workflow-Auswahl und Eigenschaften“ , S. 276*
- *Dialog „Modul-Auswahl“, S. 278*

28.14.1.1 Dialog „Workflow-Auswahl und Eigenschaften“

Ziel-Workflow

Zur Auswahl des Technical Workflows, mit dem der Workflow Connector verbunden werden soll.

In den beiden Registern werden alle lokalen und server-basierten Workflows angezeigt, für die Sie Benutzer- oder Gruppenrechte haben.

Der Workflow, den Sie verlinken, muss ein definiertes Ende haben!

Kommunikationsmodus

- **Synchron**
  Die Eingangsnachricht und die Workflow-Variablen des Workflow Connectors werden an den Subworkflow übergeben und von diesem verarbeitet.
  Wenn kein Fehler auftritt, wartet der Workflow Connector, bis das letzte Modul des Subworkflows das Ergebnis der Bearbeitung zurückgibt und setzt dann die Ausführung des aufrufenden Workflows fort.
- **Asynchron**
  Die Eingangsnachricht und die Workflow-Variablen des Workflow Connectors werden an den Subworkflow übergeben und von diesem verarbeitet.
  Gleichzeitig werden Eingangsnachricht und die Workflow-Variablen an das nachfolgende Modul im aufrufenden Workflow weitergeleitet, d. h., der aufrufende Workflow wird weiter ausgeführt und wartet nicht auf das Ergebnis des aufgerufenen Workflows.

Der asynchrone Kommunikationsmodus wird durch ein „A“ am Workflow Connector signalisiert.

→ Siehe *Symbole in Technical Workflow und im Verzeichnisbaum (Workbench: Benutzer-Guide, Kap. 12.10, S. 332).*

- **Workflow mit neuer Prozess-ID ausführen**

  (nur bei asynchroner Kommunikation)

  Wenn markiert, erhält jeder Prozess, der beim Aufruf des Subworkflows entsteht, eine neue Prozess-ID. Zur Laufzeit wird bei jedem Aufruf des Subworkflows ein neuer Prozess gestartet, der im Monitoring als eigene Prozess-Instanz geführt wird.

  Mit Hilfe der individuellen Prozess-ID können Sie den Prozess im Queue Manager identifizieren, um diesen z B. im Fehlerfall manuell neu zu starten. Bei komplexen Workflowabläufen und vielen Subprozessen können Sie die Prozess-ID nutzen, um die Prozesse zu strukturieren.

- **Fehlerbearbeitung einschalten**

  (nur bei synchroner Kommunikation)

  Wenn an einem Modul im aufgerufenen Workflow ein Fehler auftritt, hängt das weitere Vorgehen von der Konfiguration der Fehlerbehandlung ab:

  - Option ist nicht aktiviert:

    Es wird ein Eintrag im Queue Manager erzeugt. Die Fehlermeldung wird am fehlerhaften Modul angezeigt und als Inhalt der Fehlervariable `ISErrorString` an den aufrufenden Workflow Connector zurückgegeben.

    Die Verarbeitung des aufrufenden Workflows wird an dem Nachfolgemodul des Workflow Connectors fortgesetzt.

  - Option ist aktiviert:

    Der Fehler wird an den aufrufenden Workflow zurückgegeben, daher werden zwei Einträge im Queue Manager erzeugt: einer für den aufrufenden und einer für den aufgerufenen Workflow. Die Fehlermeldung wird am fehlerhaften Modul und am Workflow Connector angezeigt.

    Die Verarbeitung des aufrufenden Workflows wird nicht fortgesetzt.

- **Bei Fehlerbearbeitung Workflow-Aufruf-Stack beachten**

  (nur bei synchroner Kommunikation)

  Diese Option ist standardmäßig aktiviert.

  Wenn diese Option markiert ist und an einem Modul im aufgerufenen Workflow ein Fehler auftritt, dann wird ein Eintrag für den aufgerufenen Workflow im Queue Manager erzeugt. Der Fehler wird am fehlerhaften Modul angezeigt. Die Fehlermeldung wird nicht an den aufrufenden Workflow zurückgereicht. Der aufrufende Workflow wird nicht weiter verarbeitet.

28.14.1.2 Dialog „Modul-Auswahl“

In diesem Dialog wählen Sie das Modul aus, mit dem der Workflow Connector verbunden werden soll.

Ziel-Modul

Es wird der Workflow angezeigt, den Sie im vorherigen Dialog ausgewählt haben. Solange Sie noch kein Modul ausgewählt haben, ist das Modul ausgewählt, dessen Modulname in der alphabetischen Sortierung an erste Stelle steht.

Um das Modul festzulegen, zu dem der Workflow verbunden werden soll, klicken Sie es an oder wählen es aus der Liste über dem Diagramm aus.

28.15 Workflow Starter

Dieser Abschnitt erläutert die folgenden Themen:

- *Dialog „Workflow Starter“, S. 279*

Verwendung

Mit einem Workflow Starter können Sie Workflows im Testmodus oder zeitgesteuert starten:

- **Workflows im Testmodus starten**

  Wenn Sie einen Workflow zum Testen immer mit derselben Eingangsnachricht starten möchten, dann können Sie diese direkt im Workflow Starter speichern.

  Wenn Sie den Workflow starten, wird immer die gespeicherte Eingangsnachricht verwendet, unabhängig davon, mit welcher Option Sie den Workflow starten.

- **Workflow zeitgesteuert starten**

  Ein Workflow Starter kann, genau wie ein Systemkonnektor, Workflows zeitgesteuert starten. Zum Starten wird die Eingangsnachricht genutzt, die im Workflow Starter gespeichert ist.

  → Siehe *Dialog „Zeitgesteuerte Verarbeitung“ (Workbench/ Process Engine: Systemkonnektor-Guide, Kap. 1.1.3, S. 22).*

28.15.1 Dialog „Workflow Starter“

Statische Ausgabedatei

■ **Statische Ausgabedatei benutzen**

- Option ist markiert:

  Geben Sie eine Datei an, um den Workflow damit zu starten.

  Die Datei wird dem Workflow Starter als Kopie ohne Referenz auf das Original im Dateisystem übergeben. Änderungen der Datei im Dateisystem wirken sich daher nicht auf den Workflow Starter aus.

  Sie können entweder eine Datei aus dem Dateisystem oder aus dem Repository laden. Der Inhalt der geladene Datei kann direkt oder nachträglich durch Klick auf den entsprechenden Button angezeigt werden.

- Option ist nicht markiert:

  Der Workflow wird zeitgesteuert gestartet, wenn ein Zeitplan definiert ist, oder manuell zum Testen.

  Falls bereits eine Datei angegeben wurde, dann bleibt diese im Workflow Starter erhalten, bis sie manuell entfernt wird.